Department of Computer Science

Technical Report

Undergraduate Dissertation: Using Agent-Based Modelling to Explore the Environmental Impact of Changes to the UK Housing Stock

Liam Elliot

Technical Report 2008-03 **June 2008**

ISSN 1740-9497

Contact Address:
Department of Computer Science
University of Bath
Bath, BA2 7AY
United Kingdom
URL: http://www.cs.bath.ac.uk

ISSN 1740-9497

Using Agent-Based Modelling to Explore the Environmental Impact of Changes to the UK Housing Stock

Liam Elliott

Bachelor of Science in Computer Science with Honours
The University of Bath
May 2008

Using ABM to Explore the Environmental Impact of Changes to the UK Housing Stock

Submitted by: Liam Elliott

Declaration

This dissertation is submitted to the University of Bath in accordance with the requirements of the degree of Bachelor of Science in the Department of Computer Science. No portion of the work in this dissertation has been submitted in support of an application for any other degree or qualification of this or any other university or institution of learning. Except where specifically acknowledged, it is the work of the author.

Signed:

Using ABM to Explore the Environmental Impact of Changes to the UK Housing Stock

Using ABM to Explore the Environmental Impact of Changes to the UK Housing Stock

Abstract

With the growing public concern about global warming and reducing one's 'Carbon Footprint', household energy consumption is an increasingly prevalent topic. Given a full set of housing data, carbon emissions and energy consumption can be calculated using the BREDEM algorithm. This project was carried out in order to asses the suitability of Agent-Based Modelling (ABM) as a technique by which to model changes in the UK housing stock. The stock was modelled as a population of marionette agents (agents with no autonomous control), whose behaviour was defined by global level variables based on user-defined data. When validated in a back-cast scenario from 1996-1970, this simulation produced results within -2.65% and 2.51% of historical figures for carbon emissions and energy consumption respectively.

This accurate back-cast helped generate confidence in the forecasting ability of the model, and allowed other aspects of ABM to be explored. Several demolition scenarios were experimented with, and it was found that the choice of demolition scenario could have a bearing on the output of the model. Finally, some basic energy-related behaviours were implemented based on Van Raaij and Verhallen's *Behavioral Model of Residential Energy Use*. Although nothing conclusive stemmed from this implementation, a novel behavioural model tailored to ABM was devised and served to highlight the fact that ABM cannot fulfill its potential in this domain without further research being done into energy-related behaviours.

Contents

List of Figures

List of Tables

Acknowledgements

This project would not have been possible without the expertise and enthusiasm of Julian and Sukumar. Beyond them I would like to thank Ben, James and Trev for continually raising the bar.

Chapter 1

Introduction

"In 2004, more than a quarter of the UK's carbon dioxide emissions – a major cause of climate change – came from the energy we use to heat, light and run our homes. So it's vital to ensure that homes are built in a way that minimises the use of energy and reduces these harmful emissions." (Communities and Local Government, 2008)

In 2007, Levermore and Natarajan (2007a) developed a novel Equation-Based Model (EBM) for exploring future transformations in the UK housing stock (all of the permanent residences in the UK). Given a user-defined scenario, their method modelled how the stock changed over time. In order to build confidence in the forecasting ability of their model, their Java based implementation, DECarb, was validated using a back-cast scenario starting in 1996 and ending in 1970. In this scenario DECarb showed an average difference of -5.4% between predicted and actual energy consumption, and a difference of around -0.9% between predicted and actual carbon emissions.

Agent-Based Modelling (ABM) is a bottom-up modelling technique that traditionally aims to discover emergent global behaviours by modelling the micro behaviours of, and interactions between actors in the model, known as agents. Although it has been applied successfully in the economics and social science domains in the last ten years, the energy domain is only just beginning to take notice. Hypothetical talk of its use has been widespread for the last few years but there have been no examples of its successful application.

The two primary aims of this project are:

1. To produce a robust and extensible agent-based model of the U.K. housing stock using DECarb's existing front-end.
2. To be able to carry out validation of the model's stock transformation method using ABM against known overall energy consumption data from 1970 to 1996.

This project involved creating an ABM of the UK housing stock, building on previous work carried out by Natarajan & Levermore (2007a) on DECarb. DECarb's existing front-end allows the user to specify scenarios, for the period 1996-2050, over a range of different climate and population variables. The first objective of the project was to integrate the ABM with the DECarb front-end in order to allow a user to control the creation of the stock.

Once the components were integrated, the second objective was to create a stock transformation method in order to make the ABM population behave in the user-specified way. In order to make sure this objective was carried out properly, the model was to be validated using back-casting, a technique used with success by Natarajan & Levermore (2007a).

Several secondary objectives, to be implemented as extensions to the model if time permitted, were also specified. Implementing functionality to allow the user to choose which base year of data, 1996 or 2002, to use was one of these. Another was the implementation of a more detailed demolition sub-model. This could be used to explore demolition scenarios other than the one specified in the DECarb, which would provide an insight as to the role of demolition scenarios in the whole process.

The final two objectives were to explore the use of energy-related behaviours and the influence of government policy changes. The exploration of these would be novel within the domain, and could be hugely significant if results are found to be relevant.

This report begins with a detailed review of the literature used to inform the implementation of the ABM (chapter 2). Firstly of existing work in the domain, including DECarb and other models of the scenario, all modelled and implemented as EBM. This then leads into a discussion of ABM and the general advantages it offers over EBM, before a more specific look is taken at the use of ABM in the domain of energy research. Relevant ABM verification and validation techniques are discussed to provide an insight as to how confidence in the model is generated, and the literature survey ends with a look at technologies available to aid the implementation of ABMs.

Chapter 3 of the report details the primary aims of the project and the additional goals that were to be met if time permitted. A set of ABM design principles are derived from various points made in the literature, specifically by Inchiosa et al. (2002) and Gilbert (2004). It is then discussed how these principles led to the adoption of iterative development, and how each of these cycles were defined.

The aim of the first iteration of the project was to create a simple ABM with a population of passive agents as specified in the DECarb front-end, this is detailed in chapter 4. The first task in this cycle was to create interfaces in the existing DECarb code to allow for the integration of an ABM stock transformation method to be as smooth as possible. The high level architecture devised to allow this is discussed, as are difficulties faced in the refactoring of the DECarb code. The second stage of this iteration was to create an ABM accurately modelling the population in 1996, and the architecture, design methods, and implementation techniques chosen to carry out this task are discussed. A brief overview of creating an ABM in the framework used, RePast, is also given.

The project's second iteration, detailed in chapter 5, involved implementing decision-making capabilities in the agents, in order to allow them to achieve the distribution specified by the user in the DECarb GUI. This was achieved by treating the agents as what Gulyas (2005) calls marionettes, which involved specifying their behaviour at a global level. The propensity by which an agent may uptake an energy-related technology was defined by using formulas commonly associated with amortization. A discussion of how demolition is treated in the model is also carried out and a scalable demolition framework is described.

With a model of the UK housing stock created, chapter 6 describes how confidence in its capabilities was generated. The primary form of validation used by Natarajan & Levermore (2007a) was back-casting, effectively using the model to predict historical data dating back to 1970. The capabilities of the ABM stock transformation method were validated using the same back-casting technique, firstly docking with the figures generated by DECarb and secondly comparing them to the actual figures recorded from 1970.

Having met the primary aims of the project, chapters 7 & 8 discuss extensions carried out. A third iteration is detailed, in which the effect of using different demolition scenarios was explored. Initially demolition was a process which, given a user-defined number of dwellings to be demolished, simply demolished that number of the oldest category of dwellings. As this does not accurately encapsulate the real world scenario, 2 additional demolition scenarios were implemented and experimented with. Although one of these just demolishes a random selection of housing, the other implements a simple metric based on household insulation and aims to demolish those dwellings that are 'worst' insulated.

The final project iteration explores energy-related behavioural frameworks. The *Behavioral Model of Residential Energy Use* of Van Raaij and Verhallen (1981) is evaluated in order to find components suitable for ABM, and a simple behavioural framework is devised from this. Components in the simple framework had to meet the twin criteria of being core to the system, and also feasible to model with the existing ABM. This framework was implemented by localising information available to agents and assigning them different behavioural traits. This change in environment signified the transition from a homogeneous population of marionette agents to a heterogeneous population, of what Gulyas (2005) describes as, bounded rational agents. Several experiments were then carried out on this population of agents in order to show the possibilities that this style of model brings. Parameter sweeps are demonstrated as a technique to explore alterations to a model, and then the effect of the introduction of a government policy is looked at.

Chapter 2

Literature Survey

2.1 Domain Background

In its 2003 Energy White Paper on defining a long term strategic vision for creating a low carbon economy, the UK government set four goals, one of which was:

"to put ourselves on a path to cut the UK's carbon dioxide emissions - the main contributor to global warming - by some 60% by about 2050 with real progress by 2020" (Department For Trade and Industry 2003)

As housing is at present responsible for 26% of all UK carbon emissions (Natarajan and Levermore, 2007b), the impact of carbon emissions has been an ongoing topic for discussion in the last decade (Natarajan and Levermore, 2007a).

2.2 Existing Research

Much work has been done in an attempt to accurately model what effect different scenarios will have on the carbon emissions and energy consumption of the UK housing stock in the future, with the most notable being BREHOMES (Shorrock and Dunster, 1997), Johnston et al. (2005), the 40% House project (Boardman et al., 2005) and more recently DECarb (Natarajan and Levermore, 2007a).

All of these models have common attributes, most notably that they are physically-based, bottom-up models which work at a disaggregated level. That is to say that they start with detailed disaggregated data on physically measurable variables, such as the thermal performance of a wall or the efficiency of a heating system, and aggregate it as far as they possibly can. Each of the models output values predicting energy use and carbon emissions at some defined point in the future. Modelling in this way allows future changes to the energy saving efficiency of a dwelling (e.g. installing a more efficient heating system), as well as the impact of existing and future environmental policies and legislation to be taken into account (Johnston, 2003).

2.2.1 Stock Transformation Methods

A stock transformation method predicts future trends amongst housing categories. A Housing categories is defined by a set of attributes unique to that set of houses. To give an example; the stock transformation method will receive data specifying that in 1996 40% of dwellings have an insulated boiler and 30% have insulated walls, it will also receive user defined data specifying that this will rise to 60% and 70% respectively by 2050. The task of the stock transformation method is then to transform the base set percentages of each category of dwelling, into a new set of percentages for each category of dwelling in 2050. For this example the dwelling categories are specified in table 1:

Table 1: Example Dwelling Categories

Category	Insulated Boiler (Y/N)	Insulated Walls (Y/N)	% of dwellings in 1996	% of dwellings in 2050
1	Y	Y	X1%	
2	Y	N	X2%	
3	N	Y	X3%	
4	N	N	X4%	

2.2.2 BREHOMES and Johnston et al.

BREHOMES is a widely accepted model developed by the Building Research Establishment (BRE) which uses an average dwelling to employ the weighted average stock transformation method (Natarajan and Levermore, 2007b). Once the stock has been transformed, this information is fed into the BREDEM model which calculates the carbon emissions and energy consumption. Due to the BREHOMES model not being available to the public domain, it lacks the necessary transparency to allow us to scrutinise it any further (Johnston, 2003).

The model devised by Johnston et al. (2005) is based around the same BREDEM model that BREHOMES uses. A drawback of the Johnston et al. model is that the stock transformation method used is based on just two notional dwelling types, which it has been suggested may not lead to appropriate distributions (Natarajan and Levermore, 2007b). Again this model lacks the transparency to allow any further comment to be made.

2.2.3 Weighted Average Stock Transformation Method

The weighted average method simplifies calculations at the cost of removing a level of detail; Natarajan and Levermore (2007a) explain how this works:

"If the scenario specified the distribution of solid walls and cavity walls as 40% and 60% with 10% and 30% of total walls being insulated, respectively, then solid wall (insulated) is determined as

40% x 10% = 4% of dwellings

And similarly cavity wall (insulated) as

60% x 30% = 18% of dwelling"

They goes onto prove that this method cannot provide an accurate representation of the stock because "[it] cannot be used to accurately represent current dwelling stock since the averaging process removes the inherent variability existing in the stock distribution" (Natarajan & Levermore, 2007a).

Shorrock and Dunster (2007) claim that this technique is used due to the fact that data for future scenarios can only be found at an aggregated stock level, but Natarajan & Levermore (2007b) point out that there are other techniques available such as those demonstrated in DECarb.

2.2.4 DECarb

DECarb builds on the work done elsewhere in the domain by going beyond the weighted average method and using a more advanced stock transformation method. Natarajan and Levermore (2007a) have developed an iterative method which they claim has advantages over other weighted methods. One of its primary advantages is the fact that the dwelling categories represented at the end of the model's run are not confined to those in the base set, additional dwelling categories can be created - a case which intuitively is possible in the real world.

The Iterative Stock Transformation Method

This iterative method works in two steps.

Step One

The first step involves using a function B(x) to calculate the average change to a variable's state over a given time period. For instance the example given states:

Given a variable A with two states (0,1) and a 70-30 distribution (70% of dwellings A=0, 30% A=1)

If we were to see this distribution drop to 50-50 (50% of dwellings A=0, 50% A=1)

B(x) would return -3.33%, meaning that in the future set of dwelling categories, any category that had A = 0 would have to subtract 3.33% from its total percent of dwellings in the base year. Reversing the inputs would also indicate that any sets containing A = 1 would have to add 3.33% to its total percent of dwellings.

Further details of the algorithm used can be found in Natarajan & Levermore (2007a). This is demonstrated in tables 2 and 3.

Table 2: Base Year

Dwelling Category	**A**	B	% of dwellings
1	0	1	X1
2	1	1	X2
3	0	2	X3

Table 3: Design Distribution

Dwelling Category	**A**	B	% of total dwellings
1	0	1	X1 – 3.33%
2	1	1	X2 + 3.33%
3	0	2	X3 – 3.33%

There might also be a change in the distribution of B, creating further changes to the categories percentage of total dwellings.

Step Two

Step two is used for correcting erroneous results. If as a result of step one any dwelling categories have a representation smaller than 0%, then their representation is set to 0% and the negative value is equally divided amongst the all other positive values. When this step is complete, the sum of all percentages for a state should equal the design distribution for the state.

2.2.5 Conclusion

Although Natarajan and Levermore's stock transformation method is more extensive than transformation methods proposed before, it is felt that there is little theoretical basis for the model. The aim of this project is to create a more robust stock transformation method.

2.3 How Best to Model the Problem

When modelling a problem domain its is important to identify the individuals and observables. In the current implementation of DECarb the individuals are the different housing categories and the observables are the attributes of these categories. Using the same observables, but modelling dwellings as individuals, it is possible to take a more micro view of the problem. To this extent the problem can be viewed as a complex system.

2.3.1 Complex Systems

A complex system can be defined as a system with emergent properties that arise from non-linear interactions between its large numbers of multiple interacting constituents. Jennings (2001) defines a complex system as a many subsystems related hierarchically, these subsystems work together to achieve the functionality of their parents systems. The individual subsystems can interact with their environment and are able to respond to changes by altering their internal structure.

If a dwelling is to be the finest grain constituent of the system, it is necessary to decide exactly how one dwelling is represented.

2.3.2 Individuals as Houses

The immediately intuitive idea is to model an individual as a house. After all, it is the house that all these measurable attributes belong to, it is the house that has the insulated walls and double glazing and so on. It is the house that any energy consumption related changes will be made to. But there are also problems with this, what happens if the house is demolished? Estimations have placed an upper bound on the annual demolition rate as being 130,000 by

2030 (Lane, 2005), this would create a massive turnover every year and affect the continuity of the simulation. Additionally, a house cannot interact or exchange stimuli with another house. The argument to model individuals as houses unravels rapidly from that point onwards. It is necessary to model the physical characteristics of a house, but with behavioural properties. This leads the discussion on to the concept of households.

2.3.3 Individuals as Households

A household can be defined as the inhabitants of a house, they are not physically tied to their residence and they can move freely from one house to the next. Modelling individuals as households bridges the limitations that modelling individuals as houses provided. If a house is demolished, the household will just move to another property, households can react to changes in their environment and to information from other households. Most interestingly of all, households can have different behavioural characteristics and traits, facilitating the creation of a heterogeneous model (see section 2.4.4).

2.4 Agent-Based Modelling

2.4.1 What is ABM?

Having defined a complex system to model, the next step is to find a technique by which to do this. Agent Based Modelling (ABM) is a computational methodology whose emergence was prompted by the increased computing power of personal computers and whose popularity has increased dramatically in the last decade (Gulyas, 2005). ABM involves modelling complex systems using individual actors called Agents. Although there is no one accepted definition of an agent (Wooldridge and Jennings, 1995), Wooldridge's (2002) definition that "an agent is a computer system that is situated in some environment and that is capable of autonomous action in this environment" will be used.

Any entity, ranging from humans and animals to businesses, can be modelled as an agent, together with its imperfections, idiosyncrasies and unique interactions (Keirstead, 2005). As a result of this, intelligent agents are defined as being reactive, pro-active and social.

- Reactive. An agent is constantly monitoring its environment and has the ability to react to any changes it observes or is made aware of
- Pro-active. An agent generates and attempts to achieve a set of goals
- Social. An agent has the ability to interact with other agents

This means that every agent is capable of independent thought and actions in dynamic, non-deterministic environments (Sichman, 2007). Wooldridge and Jennings (1995) also suggest several other attributes that an agent may exhibit including veracity, benevolence, and rationality.

The social capability of interacting is defined as "communicating, coordinating and negotiating with other agents in its environment" (Wooldridge, 2002). By modelling the interactions between individual agents and their environment at a micro-level, ABM allows macro-social behaviours to emerge (Epstein, 1996). Shah and Pritchett (2004) define emergence as "as a system property in which system behaviours at a higher level of abstraction are caused by behaviours at a lower level of abstraction which could not be predicted or estimated at that lower level". Discovering emergent system behaviours is normally the goal of using ABM.

Gulyas (2005) describes ABM as "a natural representation of large dynamic populations of heterogeneous agents, which provides a way to study the system's trajectory in time". As will be shown, the fact that these agents are heterogeneous is important in modelling individuals, as when faced with the same situation, heterogeneous agents will all act differently depending on their own behaviours and plans. Therefore the power of agents is only truly utilised when they are put in an environment with other agents; a Multi-Agent System (MAS).

Wooldridge (2002) defines a MAS as containing "a number of agents which interact with one another through communication". Different agents have different spheres of influence, a part of the environment which an agent will either have control over or at least have some influence over.

2.4.2 Why Use ABM

Benefits of ABM over other modelling techniques

Bonabeau (2002) succinctly summarizes what he feels are the benefits of ABM over other modelling techniques:

- ABM captures emergent phenomena
- ABM provides a natural description of a system
- ABM is flexible

ABM captures emergent phenomena. "Emergent phenomena result from the interactions of individual entities. By definition, they cannot be reduced to the system's parts: the whole is more than the sum of its parts because of the interactions between the parts. An emergent phenomenon can have properties that are decoupled from the properties of the part" (Bonabeau, 2002).

ABM provides a natural description of a system. In many cases, ABM is the most natural medium for describing a system composed of ‘‘behavioural’’ entities. Modelling an entity’s behaviour using an agent is often much simpler than trying to devise equations to describe it. (Bonabeau, 2002).

ABM is flexible. The flexibility of ABM can be observed along multiple dimensions. For example, it is easy to add more agents to an agent-based model and to define different levels of aggregation. (Bonabeau, 2002).

ABM versus EBM

The stock transformation method in DECarb, as stated previously, is an EBM. EBM is a technique where the model is typically a set of equations and execution consists of evaluating them (Parunak et al., 1998).

ABM is not a straight alternative to EBM, Bonabeau (2002) states “a set of differential equations, each describing the dynamics of one of the system constituent units, is an agent based model”. The main factor that defines an ABM is that it describes a system from the perspective of its constituent units.

Gulyas (2005) lays-out four different implementation strategies for modelling. He then goes on to argue that the decision of the modeller is not a dichotomy between ABM and EBM, but between these four implementation strategies. The strategies move from EBM (strategy 1) to ABM (strategy 4) via 2 intermediate hybrid strategies.

1) Equation Based Formulation. Information is visible globally, the entities of the system are high level and the rules are defined at the levels of these entities.
2) Marionette. Information is visible globally, the entities of the system are modelled as individuals, but the rules are still defined at a higher level, meaning all individuals act exactly the same.
3) Autonomous agents. Information is visible globally, but rules are now defined at the individual level. Homogeneous agents in this scenario will still all act in the same way as they all have the same information.
4) Bounded rational agents. Information gets taken down to a local level alongside the rules, placing explicit bounds on the amount of information that an agent can collect from the world.

This framework blurs the line between EBM and ABM and as Gulyas (2005) says, removes the need to specifically choose one technique or the other.

Paranuk et al. (1998) compare EBM and ABM and make the following observations. There are certain similarities in EBM and ABM, at the highest level both approaches simulate a system by constructing a model and executing it on a computer. In both cases the world contains two kinds of entities: individuals and observables. Observables are measurable characteristics of interest that change over time; they can be associated with either an individual or with a collection of individuals (Parunak et al., 1998). They go on to outline two main differences between the two techniques.

The first difference is the relationships on which one focuses attention. In both techniques the observables evolve over time, but the origins of the evolution differ between the two. In EBM the modeller defines a set of equations that express relationships among observables, the evaluation of which produces the evolution of the observables. Although there may be patterns suggesting that the relationships result from the different behaviours of the individuals, these behaviours have no explicit representation, on this point, ABM differs.

In ABM the modeller defines the behaviours through which individuals interact with each other. As the model runs, close attention is paid to the observables, but throughout the simulation, relationships among the observables are an output of the process, and have evolved entirely from the behaviour of individuals.

The second difference between ABM and EBM is the level at which the model focuses. In general, some observables of interest can be defined only at the macro level (e.g. price per barrel of oil) while other can be defined either at the individual level or as an aggregate at the macro level (price of petrol at nearest petrol station vs. average petrol price in the UK). EBM makes more use of system level observables, but ABM makes more use of observables available locally to an individual as shown in Gulyas's (2005) 4th implementation strategy. Bounding an agent's knowledge in this way adds realism to the model, as it is unrealistic for an individual to have a complete and accurate view of the world.

Parunak et al. (1998) go on to say that the two approaches can be combined, as the behavioural decisions of an agent can be determined by the evaluation of some equations. System-level observables can also be introduced in an ABM, by assigning an agent the task of distributing this high level information to local agents, perhaps simulating the role that the media play in distributing news of current affairs to individuals. Gulyas (2005) agrees with this stating "how agent based a simulation is is rather a matter of degree than a binary property".

Agent Based Modelling in the Domain

Although the actual use of ABM in the energy sector to date has been very limited (Keay-Bright, 2007), recent events suggests that academics in the domain are becoming both more aware and more interested in the technology. The UK Energy Research Centre (UKERC) held a workshop in October 2007 titled *Agent Based Modelling: Application to Policy* where possible uses of ABM in the domain were discussed. The workshop opened with the speaker stating five favourable attributes of ABM compared to techniques used in more traditional models.

- Agents can have inductive knowledge. ABM recognises that people/agents are intelligent and have very good inductive reasoning (but not such good deductive reasoning).
- Traditional models are able to capture basic interactions between, for example, markets. ABM is able to explore many more dimensions of interactions such as through networks.
- ABM can represent non-linear and non-equilibrium relationships which other types of model have difficulty in representing.
- Emergence in complexity is a characteristic of ABM. For example, the macro scale will 'emerge' from the micro.
- ABM also copes with 'evolution'. Traditional models are limited e.g. with the capacity to only incorporate shocks as 'exogenous'.

These points illustrate the fact that ABM is considered suitable for the domain by its experts, although an important point brought up by one of the ABM specialists at the workshop is that there is "often an expectation that ABM is a substitute for technology, economic or market models. ABM is not a substitute, it is simply a different kind of model" (Keay-Bright, 2007). It is important that ABM is not seen as a way to optimize every existing EBM.

Although this workshop demonstrates that ABM is starting to be considered in the domain, it also demonstrates its relative youth, as one of the workshop's main goals was to bring together experts in the field of ABM and energy.

Some energy centred work that goes a step further is published in a paper by Keirstead (2005). Keirstead talks about how ABM can be used to model domestic energy consumption (DEC). A point which he feels makes ABM suitable for the domain is that "Researchers can use their disciplinary expertise to specify a particular type of agent or environment within the model". This is an important point, as Parunak et al. (1998) have proven that behaviours that can be modelled easily using agents can be "difficult to represent as rates and levels...certain behaviours are difficult to translate into [an EBM]". They go on to talk about "what-if" games, where a user only needs to define a scenario in terms they are familiar with rather than having to translate this scenario into equations relating to the observables.

Keirstead ends by saying "ABM supports an integrated approach to DEC and has many potential applications including experimentation and the development of theory". He proposes how a behavioural model devised by Van Raaij and Verhallen in 1981 can be modelled using agents which is discussed further in section 2.4.4.

2.4.3 How to Model the Problem As An ABM

Inchiosa et al. (2002) warn against two major mistakes in modelling. First of all, and perhaps most commonly, developing a model that is not well disciplined and has too many pieces.

Secondly, introducing too much complexity into a system purely for the sake of explicitly trying to model the real-world; the aim is to discover the core dynamics of the system, and not necessarily to construct a complete description. To avoid this happening they recommend starting with a very simple model which is easy to specify and implement. When this model and its dynamics are properly understood it can be extended to encompass the whole problem, an approach also favoured by Gilbert (2004).

Gilbert (2004) goes on to lay out a set of steps for modelling a multi-agent system. Inchiosa et al. (2002) add that the whole modelling process should be treated iteratively until consistency is achieved.

A research question

Start with a research question and then break this research question down into specific questions. This decomposition should stop when the specific questions reach a level of detail at which their concepts can be used as the main elements of the model.

Type of objects to be included in the simulation

The majority of these objects will be agents, but they can also include non-intelligent objects in the simulation that are used by the agents, such as the sugar repositories in the Sugarscape scenario (Epstein and Axtell, 1996). These objects can be arranged in a class hierarchy, all extending object and with the possibility of sub-classing the agents. Each object in the simulation will be an instance of a class from this class hierarchy.

Deciding Object Attributes

This works much the same way as deciding instance variables in an object-oriented class hierarchy. Attributes will be inherited by sub-classes, and can vary throughout the simulation.

Specifying the environment in which the objects are located

Without an environment an agent is effectively useless, an environment defines the properties of the world in which the agent functions (Odell et al., 2001). If the environment is a spatial one, each agent will always have a location within it, and will need an attribute to reflect this. The environment can also be represented as a 'special agent' with its own attributes and behaviours. Odell et al. (2001) specify the basic principles of a physical environment:

- Accessibility – To what extent is the environment known and available to the agent
- Diversity – How homogeneous and heterogeneous are the agents in the environment
- Controllability – To what extent can the agent modify its environment
- Volatility – How much the environment is changing while the agent is deliberating
- Temporality – Do actions occur continuously or discrete time steps or episodes

Odell et al. (2001) also specify how we must represent agent space and time with the notion of agent place, and extent and locality which are the two attributes which define place.

From static to dynamic

The model defined so far should be complete, but will be static, the next step is to allow for the dynamic behaviour. Gilbert (2002) recommends starting by considering how each agent interacts with the environment and creating lists of the actions of the agents and the environment. This will eventually lead to create a set of condition-action rules, each being associated with a unique state of the agent. It can also be common that additional attributes are required at this stage.

2.4.4 Behavioural Models

In order to specify all of the objects that are going to be in the environment, their attributes and the environment itself it can be useful to find a behavioural model to base the ABM on. This will give a theoretical basis to the work and lend more credibility to the results.

Energy Related Behaviours

Van Raaij and Verhallen (1981) define three categories of energy related behaviour:

- **Purchase-related behaviour** has to do with consideration of energy efficiency in the purchase of household appliances, heating equipment and ventilators and the relative importance and usage of the energy attribute of the products in the choice process
- **Usage-related behaviour** refers to consideration of energy efficiency in the day-to-day usage of appliances in the home. It is the frequency, duration and intensity of use
- **Maintenance related behaviour** refers to the consideration of energy efficiency in

maintaining the in-home heating system and appliances, like servicing, small repairs and small home improvements.

Barr et al. (2005) generalise these groups into two categories, habitual energy saving behaviours and purchase related behaviours. Habitual behaviours are based more around every day reductions in energy use in the house, such as thermostat settings and turning lights off. Purchase behaviours are often related long-term alterations to the property and can vary greatly in terms of cost and effort.

Factors that influence energy related behaviour

Van Raaij and Verhallen (1981) have devised a " *behavioural model of residential energy use*" that relates personal, environmental and behavioural factors to energy use. This models relationships between key aspects of household behaviour with the main aim to show how energy use of a household is influenced by energy related behaviours. Their model can be seen in figure 1.

Stern (1992) defines 10 psychological factors that act as determinants on the energy use of an individual, and defines the most important factor as the background of the individual. Learning can occur through energy bills and personal comfort levels changing attitudes and beliefs. Attitudes and beliefs can also be changed by fluctuations in the cost of fuel (Sardianou, 2007).

Consumer Characteristics

There are several important variables for defining different consumers of energy including home ownership, income/ socio-economical status, family size and age (Barr et al., 2005).

It has been shown that home ownership is the most important factor in explaining large investments energy saving measures (Black, 1985). Dilman et al. (1983) and Sardianou (2007) believe that household income is a dominant predictor of energy use behaviours and they found that people on lower incomes were more likely to make changes in their behaviour to save energy, while those on higher incomes were more likely to invest money in changes to their property. Ritchie et al. (1981) found that there was a strong positive correlation between the age of the head of the house and general energy saving, although Sardianou (2007) found a positive correlation between age and energy consumption.

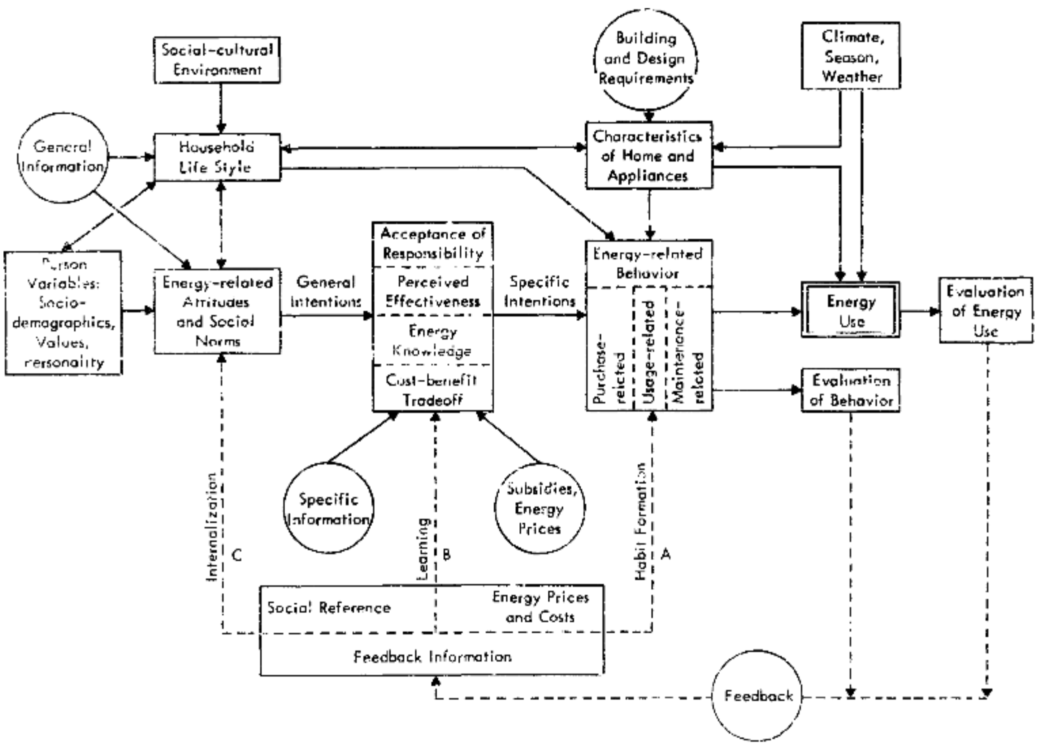

Figure 1: Van Raaij and Verhallen's Behavioural Model of Residential Energy Use

Many different papers have attempted to categorize types of energy consumers. Sardianou (2007) carried out empirical research on the Greek population and came to the conclusion that "consumers who have higher private incomes, own their houses and are members of an extended family core are more likely to make a conservation improvement". He goes on to say that sex, marital status and education cannot be used as factors to predict energy consumption, and the number of rooms in and size of a dwelling do not seem to affect what kind of energy saving strategies are undertaken.

Barr et al. (2005) define four different types of energy consumers based on behaviour; Committed Environmentalists, Mainstream Environmentalists, Occasional Environmentalists and Non-Environmentalists. In terms of purchase behaviour the first two categories are effectively the same so they can be amalgamated. The characteristics are much as to be expected, with the more environmentally friendly consumers living in smaller properties that they are more likely to own outright. The less environmentally friendly consumers were found to have larger homes that they were less likely to own. Consumers who took part in no energy saving activities at all were most likely to have the lowest income, although, surprisingly, they were also the most techno-centric in outlook.

From a Behavioural Model to an ABM

Using a behavioural model for ABM is a concept proposed by Keirstead (2005), who talks about "taking the comprehensive outline of Van Raaij and Verhallen and reordering the multiple constituents of DEC into tangible agents". He further demonstrated these ideas with two high level sketches which can be seen in figures 2 and 3. Bearing in mind the concepts proposed by Inchiosa et al. (2002), the sketches in these figures appear to be attempting to model too much of the problem, and could do better to attempt to model the core components first. Attempting to model every facet of Van Raaij and Verhallen's model could be seen as introducing complexity just for the sake of explicitly modelling the real world dynamics.

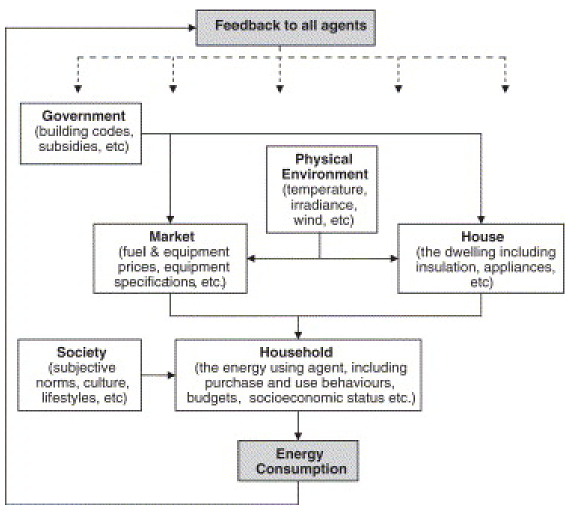

Figure 2: ABM framework for DEC

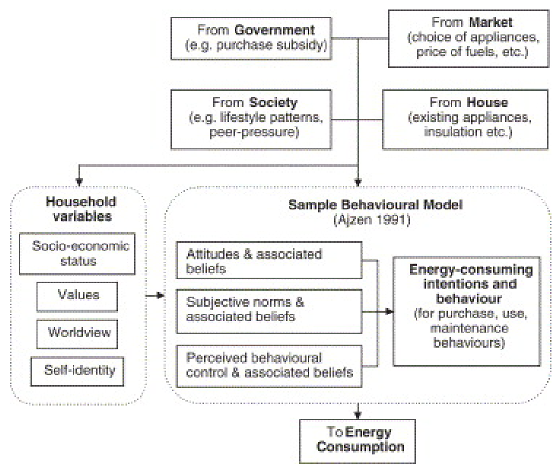

Figure 3: Household agent for DEC

2.5 Validation & Verification of ABM

Although validation and verification are commonly mentioned in the same breath, it is important to define the difference between them. In the very simplest terms verification can be described as asking the question "Did I produce the simulation right?", whilst validation can be described as asking the question "Did I produce the right simulation?" (Shillingford, 2005). Kennedy et al. (2005) elaborate on these terms by saying that verification involves making sure that the code generating the phenomenon has been modelled correctly to match the abstract model, while validation is making sure that the correct abstract model was chosen to accurately represent the real-world phenomenon. They go on to specify that it is important to perform these two tasks in tandem to obtain the best results.

One of the important tasks for a simulation study is determining how accurate a simulation model is with respect to the real system (Xiang, 2005). Effective validation and verification can increase confidence in the model (Kennedy, 2005), which in turn makes the outputs more informative and valuable.

Balci (1998) outlined 15 general simulation principles to help modellers better understand the verification and validation that they are performing. The two most relevant principles for ABM are:

- The outcome of the model verification, validation and testing should not be considered as a binary number.
- Complete simulation model testing is not possible. As not all inputs and parameters can be tested, only the most appropriate ones should be chosen.

The first point emphasises the fact that ABM will often simulate scenarios with no 'right' or 'wrong' answer, and so aspects of the situation can be modelled correctly, whilst other aspects may require fine tuning, further demonstrating the need for iterative development. The second point emphasises that focusing on the core components of the system is essential, they are the components that will have the greatest bearing on the outcome of the simulation and therefore should be tested more vigorously than other, less key, components.

He goes on to state that verification and validation can be broken up into two stages, subjective methods and quantitative methods.

2.5.1 Subjective Methods

This set of methods largely relies on the judgement of domain experts, and is often used for initial “quick-and-dirty” validation. Although the techniques are not as formalised as the quantitative methods, subjective methods can be used to detect flaws early in the simulation process (Kennedy, 2005). There are several different subjective methods to consider:

Face Validation

This is a very early phase of validation that involves domain experts judging whether the model behaves in a reasonable manner. Initially this can be carried out on a subset of the model's functionality in order avoid the cost of running full simulations.

Some ABM toolkits contain features which aid face validation such as animation or visualisations/ graphical representations. Animation is the graphical display of the behaviour of the model over time. Visualisation involves representing the model’s output data with various graphs, these graphs can aid the domain specialists in their subjective judgement of the simulation (Xiang, 2005).

Tracing

This is a technique similar to animation and involves following the behaviour of any number of entities in the model to determine if the behavioural logic in the model is correct (Kennedy, 2005). Tracing can be carried out graphically, using graphs for example, or carried out as a more numerical process by analysing the data output by a model.

Turing Test

A Turing test in this situation would involve asking a domain specialist to see if he could tell the difference between output from the new model and the existing model. If the outputs are so similar that the specialist could not tell the difference then the test has been passed (Kennedy, 2005).

Parameter Sweep

The parameter sweep is a software infrastructure for the systematic execution of experiments across large regions of parameter space. It allows the user to observe how the model acts over the whole range of values for any parameter and to collect and archive observations of the system's operation in specified metrics (Brueckner, 2003). It can be used in the verification process to find the parameter settings that produce the best output. Conducting a sweep across all parameters is not feasible in the majority of simulations (Robinson, 2007), and therefore it is important to sweep only those parameters which it is felt will make a noticeable difference to the results.

2.5.2 Quantitative Methods

Quantitative methods facilitate model validation by using statistical techniques to compare model output data with trusted sources.

Model-to-model comparison/ Docking

Docking is a technique that compares various results of the simulation model to results of other models, both take the same input and subsequent differences in the output reveal problems with the model accuracy. If docking an unproven model with a valid model, an agreement between the two models can infer the validity of the unproven model (Xiang, 2005).

"By comparing simulations built independently using different simulation tools, the docking or alignment process may discover bugs, misinterpretations of model specifications and inherent differences" (Xu, 2003).

After a set of appropriate output measures have been determined, statistical testing can be carried out to compare results between systems.

Historical Data Validation

This involves validating the model against historical data. Natarajan & Levermore (2007a) use a technique called back-casting, where the scenario specified is one that has happened in the past, in their case between the starting year of 1996 and 1970, which they describe as working by:

"comparing the results of the back-cast with known energy consumption data. Back-casting in the model is very similar to forecasting with the difference that forecast generates new dwellings in addition to simulating change in the numbers of existing dwellings, whereas back-casting involves only the latter. Since the modelling of new dwellings involves exactly the same methods and objects as the modelling of existing dwellings, the accuracy of the back-cast can be used as an estimate of the confidence in a forecast."

Therefore the data produced by back-casting has some concrete historical data against which it can be compared, and if this comparison produces favourable results it will improve the credibility of any forecasts.

2.6 Toolkits/ Frameworks

"Simulation toolkits ease the burden of computational modelling by offering a set of building blocks and standardized solutions to frequently occurring tasks and problems" (Gulyas, 2005). Keirstead (2005) and Parker (2001) both feel that frameworks assist modellers by allowing them to focus on the agent, its behaviours and the basic interactions driving the model. This will save them from getting side-tracked by trivial issues such as model execution and visualization.

Parker (2001) states that there are several considerations which often hold modellers back from using frameworks, including:

- ABM frameworks often appear daunting and the benefit of such tools does not seem to warrant the cost of learning them
- Software developers and researchers are culturally very sensitive to loss of control and are not comfortable with subjecting important parts of models to a 'black box' implementation

There are a variety of different toolkits available and in choosing one there is often a trade off between flexibility and ease of use (Gulyas, 2005), some toolkits are designed to support modelling over a wide variety of domains, while others are designed with a specific domain in mind and offer functionality accordingly.

2.6.1 Swarm

Swarm is a software package for multi-agent simulations of complex systems (Xu, 2003), and was created with the aim of removing some of the low-level complexities found in ABM at the time.

Before the birth of Swarm, ABM could be carried out at the lowest level with C, but this required the modeller to consider additional issues such as memory management and time steps. Using C++ or Objective-C removed the memory management issues, but time steps still needed to be run via many loops. SWARM was introduced as a high-level environment designed to remove time steps from the modeller's thinking (Terna, 1998).

Swarm was based on Objective-C and was the first ever package for ABM (Gulyas, 2005), it contained a standardized set of tools, all of which were usable on a wide variety of systems. Objective-C is a rarely used, object-oriented take on C, lacking strong typing, and as such, Swarm has a steep learning curve.

More recently the makers of Swarm released a successor to Swarm called JavaSwarm, this involved a Java layer running on top of the Swarm kernel. Although JavaSwarm was written in Java, the underlying use of the Swarm kernel meant that it still had an Objective-C feel about it. Many of Swarm's concepts and design patterns are still fundamental to most ABM toolkits (Gulyas, 2005), (Collier, 2002).

2.6.2 Ascape

Ascape is a second generation based simulation toolkit written in Java, but with a design modelled after Swarm (Gulyas, 2005). It aims to be expressive, by making it possible to "define a complete model in the smallest possible description" as well as providing high level user oriented tools that allow users to model interactions without any programming at all. Another of the design goals of Ascape was to "encapsulate basic agent modelling ideas [...] while providing a great deal of flexibility and expressive power" (Parker, 2001). Gulyas (2005) disagrees with the level of flexibility offered by claiming although Ascape has powerful support for cellular automata-like models, this introduces a rigidity which makes it very hard to implement other kinds of models.

2.6.3 RePast

The makers of RePast originally set out to create something similar to JavaSwarm, but as the makers of Swarm got there first, they decided to develop an independent framework written entirely in Java (Collier, 2002).

RePast is a software framework for agent-based simulation created, by Social Science Research Computing at the University of Chicago, specifically for ABM in the social sciences (Tesfatsion and Axelrod, 2007). It provides an integrated library of classes for creating, running, displaying, and collecting data from an agent-based simulation. Although RePast is an independent framework it does still borrow much from the design of Swarm and as such can be termed a "Swarm-like" simulation framework (Collier, 2002).

RePast has an unconstrained approach – allowing any type of agent based model, and also offers explicit support for several common ABM tasks (Gulyas, 2005). In addition to this, two of RePast's design goals were a short learning curve, and an acceptable level of performance when weighed against other benefits of the toolkit. The short learning curve is a plus point for RePast and although an acceptable level of performance does not sound so impressive, RePast still offers comparable performance to similar frameworks (Collier, 2002). RePast also provides a wide range of library packages which allow the modeller to access features such as QuickTime movies and snapshots (Xu, 2003).

RePast uses Java and a stated benefit of this is that it eliminates memory leaks (found in C, C ++ and Objective – C) which are often problems for long running simulations (Collier, 2002). The use of Java does require the modeller to have some experience using the language, but the fact that the toolkit has been widely used across the Department of Computer Science offers further support for RePast's case.

2.6.4 Using a Rules Management System

Using a rules management system for the agent's decision making has the advantage of logic and data de-coupling which will increase the maintainability and logical flow of the code (JBoss Rules, 2007). Robinson (2007) assesses the suitability of several rules managements systems, and the two most widely used, stable options are JBoss Rules and JESS:

- JBoss Rules or Drools as it is more commonly known, is an implementation of the Rete algorithm in Java. It uses the concept of working memory, in which a set of facts are stored. These facts are continually checked against the user defined rules and if any of these rules are met then the rule is fired and an action is run. Facts are cached in memory until they change, meaning less computation is necessary. Drools is an established piece of open-source software with a simple minimalistic syntax. There is experience of using Drools in the Department of Computer Science.

- The Java Expert System Shell (JESS) is also a Java implementation of the Rete algorithm, but uses a Lisp-like syntax. Like Drools, Jess has an established support base, but is not open-source – although it is freely available for academic use.

2.6.5 Toolkit of Choice

To choose a toolkit for this project, Parker's statements were taken into consideration (see section 2.6). The second point can be met by any of the proposed toolkits as they are all open source and therefore should not be considered as black-boxes.

In the case of RePast, it appears that the benefits that would be gained from using it far outweigh the costs occurred from learning to use the it. Features are provided that it would not be time-effective to develop independently. These costs are further softened by the depth of departmental experience in using it.

The same applies for the choice of Rules Management System, as the simpler syntax and departmental knowledge of Drools appears a more attractive choice than JESS.

2.6.6 Random versus Pseudo-Random

Random number generators cannot be created to produce *entirely* random numbers. Random number generators actually produce pseudo-random numbers. That is to say, given the same initial seed, random number generators will always produce the same sequence of outputs. In the Java API there is a library providing a random number generator which is an implementation of the linear congruence algorithm (Java API). As far back as 1968, mathematicians such as Marsaglia (1968) were aware of several flaws in this algorithm that caused the generated numbers to not be distributed as evenly as would be hoped.

A benefit offered by RePast is the use of the Colt random libraries. These libraries provide random number generation by using an implementation of one of the strongest pseudo-random number generators, the Mersenne Twister (Collier, 2002). Rabone (2005) compares the standard Java random number generator with an implementation of the Mersenne Twister in Java. He found "severe non-randomness & periodicity" in the Java generator and when comparing the two using the DIEHARD tests found that the Java generator fails several of the tests. Despite all of this, Rabone found the Mersenne Twister to be quicker than the Java generator.

2.6.7 Creating an ABM in RePast

Creating a basic model in RePast is very straightforward due to numerous online tutorials detailing the process (e.g. Tesfatsion, 2008).

Murphy (2004) lists the three main components of the ABM in RePast as; the Model, the Space and the Agent. The model object acts as the actual model and the space object controls the environment in which the agents interact and co-exist. Although the model is the most complicated part of the simulation, a lot of what goes on is hidden from the programmer by sub-classing any new model from an existing class, SimModelImpl. A similar situation occurs with the space object which can just contain a standard RePast container for the agents.

2.7 Conclusion

This literature survey has looked at work from each of the relevant domains and limited the scope of the project by determining which research has already been carried out and which areas will need to be looked at in more detail.

It has been seen that although there are several existing and well respected models which predict future carbon emissions and energy consumption, they all use fairly basic stock transformation methods with no theoretical basis.

In order to find a more robust stock transformation method for DECarb the possibility of modelling the UK housing stock as a complex system using ABM has been explored. The advantages of using ABM over EBM for this kind of problem have been discussed, and several behavioural models of energy consumption upon which the system could be based have been found.

The survey ended with a discussion of ABM toolkits and rule management systems available, and their relative merits. RePast was chosen as a suitable toolkit with which to create the ABM and JBoss Rules was selected as the rule management system.

Chapter 3

Requirements Analysis and Specification

3.1 Requirements Elicitation

This project was proposed by Dr. Sukumar Natarajan – co-creator of DECarb – and it was necessary to meet with him at an early stage to elicit the initial requirements. Upon first meeting Natarajan, a set of requirements were agreed, specifying the aims of the project. Two, high-level, concrete aims were agreed upon:

1. **To produce a robust and extensible agent-based model of the U.K. housing stock using DECarb's existing front-end.**
2. **To be able to carry out validation of the model's stock transformation method using ABM against known overall energy consumption data from 1970 to 1996.**

Four additional goals – of equal priority – were also agreed upon. These were not considered vital for the success of the project, but interesting enough for consideration if time permitted.

1. To be able to **choose the base year** from which the model works.
2. To implement **behaviours** within the individual agents, creating a heterogeneous society.
3. To model the impact of **policy measures**, on a household's behaviour.
4. To create a **demolition sub-model** in order to further explore the effect of demolition**.**

3.2 Design Principles

In addition to the functional requirements specified in 3.1, the literature survey provided several design principles to bear in mind when carrying out a project using ABM. Of particular interest is section 2.4.3. From recommendations proposed by Inchiosa et al. (2002) and Gilbert (2004), the following design principles were derived:

1. Model the problem iteratively
2. Start with a simple model which is easy to specify and implement
    - 2.1. Begin with a set of passive agents which do not interact with their environment
    - 2.2. Once this model and its dynamics are properly understood, extend it to encompass the whole problem
3. Model only the core dynamics of the system, do not try and explicitly model the real world
    - 3.1. Only go to a necessary level of detail, do not add complexity for complexity's sake

The most important point taken from these principles is the need to stress simplicity at every stage of the project. Iterative development is a very important technique in this quest, as it allows the developer to consider problems on a smaller scale, and not get bogged down in potential complexity.

It was decided that some of the software development principles of the Agile Manifesto would ideally lend themselves to these ABM design principles

3.2.1 The Agile Manifesto

The original Agile Manifesto, as agreed on by 17 experts, contains 12 principles to aid 'better software development' (Agile Alliance, 2001). Not all principles were found to be applicable to this project, but these three were particularly relevant:

Simplicity--the art of maximizing the amount of work not done--is essential

This principle concurs with the ABM specific principles proposed by Gilbert (2004) and Inchiosa et al. (2002), and the project needed constant re-evaluation against it, as the scope for complexity when modelling the UK housing stock is enormous.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale

This principle was achieved by, as mentioned, carrying out iterative development. Cockburn (2002) expands on this point by specifying that although delivery cycles should rarely be shorter than one month, it is still important to liaise with the user(s) mid-cycle in order to get useful feedback, which leads onto the third and final principle.

Business people and developers must work together daily throughout the project

Following this principle strictly would be counter-productive on a project of this size, but the underlying value is still applicable. Namely, constantly fine-tuning requirements with and getting feedback from Natarajan was vital in-order to successfully meet his aims.

3.3 Non-Functional Requirements

In addition to those functional requirements specified by Natarajan, a list of non-functional requirements was also drawn up.

1. An Integrated Development Environment (IDE) should be used
    1.1. The IDE should be compatible with RePast
    1.2. The IDE should be compatible with JBoss Rules
    1.3. The IDE should have a reasonable sized support community
    1.4. The IDE should have adequate documentation
2. Version Control Software (VCS) should be used on a regular basis
3. The simulation should be capable of running on a standard specification machine with 1gigabyte of RAM
4. The ABM should be implemented in a language compatible with RePast and with DECarb

These requirements were derived partly from the developer's own experiences and partly from the conditions under which the project took place. Use of an IDE with the specified features was a requirement created in order to increase the ease of the implementation process as much as impossible. Using VCS was deemed essential not only in case of hardware failure, but also to provide access to past versions of code if changes had been made inadvertently. Requirement 3 was defined by the hardware available, and was necessary in order to scope the project realistically. The final requirement guiding the choice of implementation language was essential in order to end up with a working system.

3.4 Iterative Delivery

In order to formalise the iterative delivery process, it was necessary to define iterative stages of development, each having its own set of objectives, the details of which are specified in each section. A high-level overview of the aim of each cycle is given.

3.4.1 Cycle 1

The two primary goals of cycle 1 were firstly to create clear interfaces in the existing DECarb code in order to allow for the clear flow of data in and out of the ABM, and secondly to create an ABM of passive agents, representing the UK housing stock.

In order to have an exact idea of the potential inputs to the ABM, refactoring the DECarb code was chosen as the first task to be carried out. Once equipped with knowledge of the necessary inputs and outputs, the process of creating a system of passive agents before implementing additional functionality is consistent with the design principles proposed in section 3.2.

3.4.2 Cycle 2

The aim of cycle 2 was to devise and implement a technique to make the agents act in such a way that the user defined uptake levels were met for all household attributes. A successful completion of cycle 2 would meet the first aim of the project.

Given an accurate system of passive agents, the bulk of the task of cycle 2 was to choose a suitable technique by which to make these agents behave in the required manner. An entire cycle was devoted to this search due to the fact the solution would have to be devised from scratch.

3.4.3 Back-Casting

The aim of the back-casting phase was to generate confidence in the ABM. The back-casting technique carried out by Natarajan & Levermore (2007a) was the method of choice to compare the results obtained from the ABM against both real figures and DECarb's output for the same period. Successful completion of this testing would ensure that the two primary aims of the project were met.

3.4.4 Discussion of additional functionality

After the completion of testing, both of the primary aims should have been met. Therefore each of the additional goals were analysed for feasibility and usefulness, in order to define the aims of any additional work.

1. To be able to **choose the base year** from which the model works.

DECarb generates its base population using the housing stock data from 1996, similar data exists for 2002. A useful additional piece of functionality would be to allow the user to choose which base year to create the ABM from.

2. To implement **behaviours** within the individual agents, creating a heterogeneous society.

This is an area which, as demonstrated in the literature survey section 2.4.4, a fair amount of empirical research has been carried out on in an attempt to categorize householder's energy related behaviours. Unfortunately, there has been little extensive research done on energy-related purchasing behaviours. If there existed a model which placed every household into some energy purchasing behavioural category, it would lend itself well to an ABM scenario, where every agent could be assigned a behavioural trait.

3. To model the impact of **policy measures**, on household's energy-related purchasing behaviour.

Governmental policy measures are something which, in the real world. undoubtedly have an effect on people's energy-related purchasing.

The U.K. government has recently released a **Code For Sustainable Homes**, which defines minimum standards for new dwellings and gives homes a rating depending on their level of sustainability (Communities and Local Government, 2008). Exploring the impact of this code on the environment would be very interesting, but possibly problematic. One facet to this problem is that to calculate the carbon output of a household, the current data for that household would have to be processed by DECarb's energy calculator. This would greatly increase the duration of the simulation and place additional computational strain on the hardware on which the simulation was been run.

The government also run the **Warm Front Scheme ,** which offers grants for heating and/or insulation upgrades to a variety of deserving applicants who could not otherwise afford it (The Warm Front Team, 2007). Modelling this would first of all involve assigning an income band attribute to every household, and then introducing some kind of behavioural or awareness factor, as it is unlikely that everyone with the opportunity to take up this grant actually does.

With the correct empirical research in place, these schemes would be feasible options to model, but they are not areas in which a huge amounts of research has been done, and any agent decision-making based on this would have no theoretical basis. That is not to say that either of these cannot be added to the model at a later date, as the flexibility of the ABM would easily allow this.

4. To create a **demolition sub-model.**

A demolition sub-model would allow for more experimentation as to how demolition is modelled. The DECarb GUI allows the user to specify the demolition rate (houses demolished p/a) for every year leading up to 2050. The current DECarb EBM just demolishes the houses in the oldest age class first, and then when all of the houses in that age class have been demolished, moves onto the next.

This is, of course, not a strictly realistic model of the real world, and looking at modelling this differently could provide more accurate results. Discussions with Natarajan brought up three different scenarios that could be explored:

- The oldest properties are demolished (as existing system)
- Random properties are picked for demolition
- Using some kind of metric, properties with the worst insulation are picked for demolition

As well as providing possibly very interesting results, this is the kind of situation which ABM excels in, and offers an additional level of flexibility over EBM. Modelling every household as an individual agent means that as long as a list of all agents is stored somewhere, it is no more complex a task to pick a random set of agents, than it is to pick the oldest set of agents.

Conclusions

Although changing the base year of the simulation would be an interesting task to carry out from an energy point of view, it is a trivial task from a complexity point of view, so this goal will not be explored in this project.

Exploring different demolition scenarios is an excellent example of how best to make use of flexibility offered by ABM, and has the potential to produce some thought provoking results regarding the impact of demolition scenarios.

It was felt that the other 2 goals could be explored in a similar vein of work. Implementing a simplified behavioural framework, could then lead on to experiments demonstrating the impact that government policy changes can make. These goals were constrained by the lack of literature in the domain, and so were considered last.

3.4.5 Cycle 3

The aim of cycle 3 was to further explore demolition scenarios and their effect on energy consumption and carbon emissions, by implementing the three scenarios discussed with Natarajan.

The rationale behind experimenting with different demolition scenarios, is that it could help to determine the overall effect of the chosen demolition scenario on the output of the model.

3.4.6 Cycle 4

The aim of cycle 4 was to explore the effect of introducing energy-related behaviour to households by devising and implementing a simple energy-related behavioural model. Once the model was implemented, the aim was to explore how the effect of governmental policy changes could be modelled.

Modelling the energy-related behaviours of households is a domain large enough to warrant its own project, so the aim of creating a simple behavioural model was primarily a task in showing what *could* be achieved.

Chapter 4

Cycle 1 – Creating the ABM

As specified in the previous section, this project was carried out in an iterative fashion. That is to say, no aspect of cycle 2 took place until cycle 1 was marked as complete. For this reason, each cycle will be described in full, from requirements to implementation, before the next is begun, this approach aims to give not only a more chronological view of the project but a more logical one.

4.1 Requirements For the First Cycle

The aim of the first cycle was to produce a model of passive agents representing the households in the UK housing stock, in which they were aware of their existence, but did not interact in any way with each other or their environment.

1. Define clear interfaces in the code between ABM and DECarb in order to ensure loose coupling between the two.
2. Given the user inputs from the DECarb GUI, create the ABM
    - 2.1. Model every household as an agent
        - 2.1.1. Assign each household the physical characteristics specified by the user (i.e. wall insulation, window insulation etc.)
    - 2.2. Make sure that the agents are aware of themselves and their neighbours
3. For each of the marker years (1996, 2000 – 2050) record summary data to allow energy calculations to be carried out
    - 3.1. Data should be sufficient to allow DECarb to calculate energy consumption in 2050

Stressing simplicity at every stage was one of the design principles of this project, and the rationale behind first creating a system of passive agents was to ensure that no unneeded functionality or detail was implemented. Rather than design the ABM as an entirely separate component and wait until the end of the project before integrating with DECarb, early integration was specified as an important requirement. The rationale behind integrating first was that refactoring DECarb would acquaint the developer with its structure and functionality, which would be essential in order to ensure smooth integration.

Before going into the details of the design, it is prudent to explain exactly what the user can specify using DECarb and how the DECarb front-end should interact with the ABM module.

4.2 Existing DECarb Front-End

DECarb allows a user to specify a very detailed scenario regarding the housing stock, and observe predicted carbon emissions and energy consumption for the years 2000, 2010, 2020, 2030, 2040 and 2050 if this scenario comes to pass.

Using DECarb for the first time is a daunting task. One is presented with a myriad of graphs and figures which, to someone with no experience in the domain, can appear to make very little sense (see Appendix A).

It is not necessary to give the reader a full overview, as many of the tasks that the user can carry out using the GUI have no bearing on the modelling of the housing stock, and are used just for energy calculation. An overview will be given explaining how the values that are used in the ABM are defined in the GUI.

4.2.1 Specifying How Household Characteristics Change Over Time

The input method for the majority of a household's characteristics is a malleable graph, as seen in figure 4. In this example, the characteristic is window insulation. With this graph, the user can define what fraction of the population will have single-glazing and what fraction of the population will have double-glazing at each year. The user has specified that, in the year 2030, 93% of the population should have double-glazed windows.

Window insulation is a simple example as every agent merely stores a binary value to represent it. For attributes such as; heating system, construction type and dwelling type there are multiple values, for example a household can have one of six different heating systems. As such. the input graphs for these are slightly more complicated than figure 4, but are still intuitive to use.

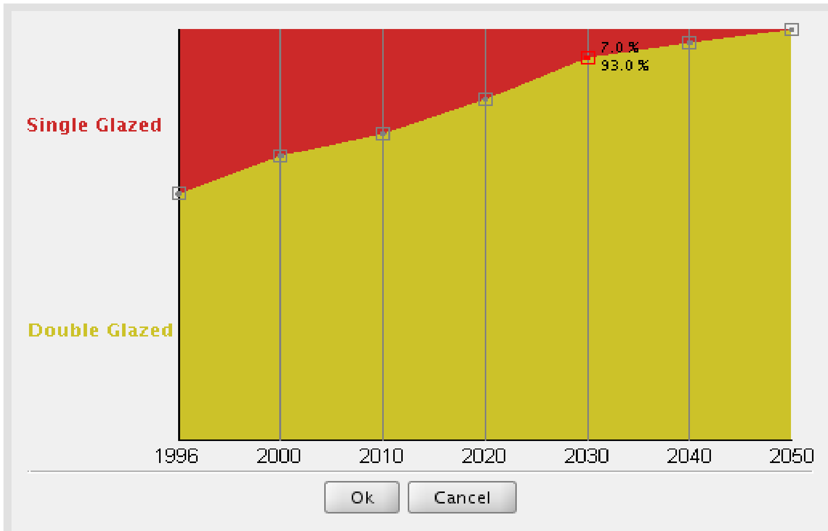

Figure 4: Input Graph for Window Insulation in DECarb

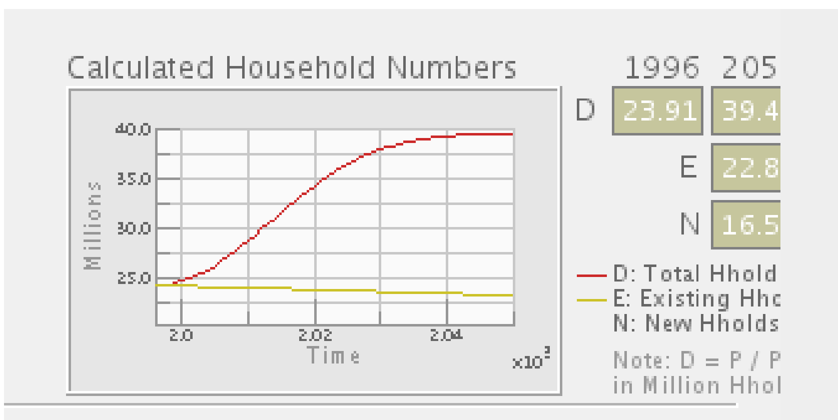

Figure 5: DECarb Calculated Household Numbers

Apart from these house attributes, the only other functions in the GUI which are of use to the ABM are those regarding population size. DECarb allows a user to specify what they think the population, people per dwelling, and demolition rate will be for each year leading up to 2050. From the user's manipulation of these three malleable graphs, DECarb calculates the total population for each year, and breaks this down into those dwellings constructed prior to 1996 (existing households*)* and those dwellings constructed after 1996 (new households*)* as shown in figure 5. The styles of graphs demonstrated in figures 4 and 5 are the only two available in the DECarb GUI. All graphs either allow the user to define a fraction of the population, as in figure 4, or allow the user to map a linear relation between some value and time, as in figure 5.

4.2.2 Age Classes

Another recurring theme that requires definition is that of age classes. An age class is a group of households built in a certain period of time, more formally it is a surjective mapping from the set of households to the set of age classes. The dwelling data used to define the housing stock in 1996 is split into 6 age classes, each of which represents a different period of pre-1996 housing and as such the number of households in these classes does not increase during the simulation. At each period in time for which DECarb records data, a new age class is started, so dwellings built between 1996 and 2000 fall into age class 7, this is shown in table 4.

Table 4: Relationship between Age Class and Year of Construction

Age Class	Built in Period
1-6	6 Sequential Periods Pre-1996
7	1996-2000
8	2001-2010
9	2011-2020
10	2021-2030
11	2031-2040
12	2041-2050

4.3 System Architecture

When thinking about how data currently flows through DECarb, it is simple to visualise it as three components. The GUI front-end, the EBM used to model the stock transformation in the middle, and then the DECarb energy calculator situated at the back-end, as shown in figure 6.

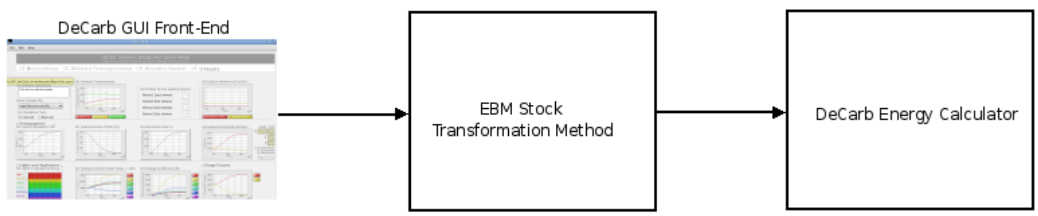

Figure 6: Existing DECarb data flow

From the data flow perspective it should have been a simple case of re-routing data from the front-end through our ABM module, rather than using the existing EBM module, as shown in figure 7. This could have been done by refactoring the existing EBM module to implement an interface, and then creating the ABM to implement the same interface, making it very easy to swap between the two.

Figure 7: Proposed DECarb data flow with ABM

Unfortunately, DECarb was not programmed in an object-oriented manner and there was very tight coupling between each of the components, with the majority of the work taking place in a couple of very large classes.

For this reason, implementing a common interface was not immediately feasible, and some refactoring of the DECarb code was necessary.

4.4 Refactoring DECarb

The aim of this early refactoring process was to get a trivial, but time-consuming, task out of the way so that the ABM would be supplied with correct data from the start. An additional design objective was to keep the ABM module as a standalone component, effectively a library class, that could be utilised by DECarb, the benefits of this being that any changes to DECarb would not affect the ABM and that it would allow for easier testing.

There were several different ways that the DECarb code could have been refactored. One option would have been to move the EBM code into another class and implement an interface that the ABM could also have implemented. This would have been a very tidy way to solve the problem and would have allowed for easy docking between the two modules. The main drawback of this solution was that it would have involved moving around a lot of existing DECarb code, although, given more time, this would not have been a massive problem, it was decided that the existing DECarb code should be left alone as much as possible.

So, rather than refactor the DECarb code to allow for the ABM, it was decided to use the Adapter design pattern instead.

"Adapters are used to enable objects with different interfaces to communicate with each other" (Gamma et al., 1994)

The adapter, as seen in figure 8, sits in-between DECarb and the ABM and is tailored to work with their interfaces. Therefore in the DECarb code, all of the data from the front-end is sent to the adapter and then the adapter is told to begin the simulation. This has the additional bonus of hiding the complexity of the ABM from DECarb and ensuring there are minimal interactions between the two.

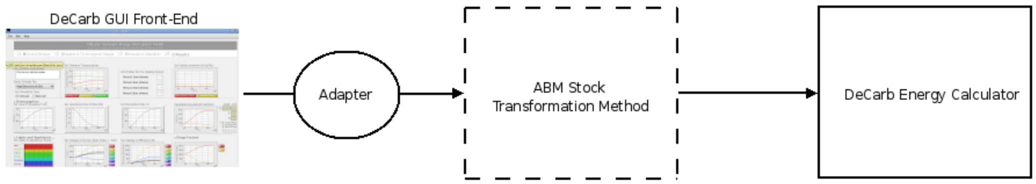

Figure 8: Proposed DECarb Architecture with Adapter

Although not explored in this project, this architecture would also allow for unit tests to be written for DECarb where a 'mock' adapter interface could return some constant results. This would mean DECarb's output from then on would be deterministic and therefore easily testable.

Once the user-defined agent data was obtained successfully from the adapter, the next stage was to create the model of passive agents.

4.5 Choice of Implementation Language

The choice of Java as the language of implementation was a logical decision from a quick observation of the situation. One of the primary aims was to integrate an ABM stock transformation method with the existing implementation of DECarb. The fact that DECarb was implemented in Java, meant that any other choice of language would have provided very costly time overheads during the integration process. The author's experience in Java, as well as the existence of a Java implementation of the toolkit of choice – RePast (see 2.6.5) – removed any doubt as to whether Java would be the correct language to implement in. Although, it should not be inferred that the selection of Java was a forced choice, it has many advantageous attributes.

Java is an object-oriented language, that is to say that it uses objects and offers dynamic lookup, abstraction, subtyping, and inheritance. Object-oriented languages are ideally suited for modelling real-world problems, as real world concepts and entities can each be embodied as individual objects.

The portability, or platform independence, of Java meant that the ABM could be developed and tested on any platform. This may not be hugely important in a standard development process, but it was re-assuring to have a redundant option if implementation on the chosen platform ever became infeasible.

Simulations of a large size can place place a huge strain on the hardware they are being run on, and any savings that can be made are essential. Another feature offered by Java is dynamic linking, that is the functionality of loading classes into the Java Virtual Machine

(JVM) incrementally, only as and when they are used. Also of interest here is the feature of Java garbage collection. Java automatically determines whether memory can be de-allocated at run time. (Mitchell, 2003)

4.6 Development

This whole project was carried out on a laptop with 1 gigabyte of RAM and an Intel Core 2 Duo Processor. The laptop was running 64-bit Ubuntu 7.10, and this enforced an additional requirement that all technologies used must be available on the Linux platform, instantly ruling out the entire range of Microsoft offerings.

4.6.1 Choice of IDE

The choice of free, stable, IDEs with user support bases large enough to offer help is sparse. The two main options appeared to be NetBeans and Eclipse. Although NetBeans 6.0 had the advantage of offering a built-in profiler as well as many of the same features of Eclipse, Eclipse was the IDE of choice.

This was mainly due to the availability of both RePast and JBoss Rules plugins for Eclipse. The RePast plugin allowed for RePast to be integrated with Eclipse and to run RePast models from within the IDE. The JBoss Rules plugin offered the advantage of syntax highlighting for .drl files, and allowed the programmer to import JBoss Rules libraries into a Java file, making development very easy.

Although not offering a built-in profiler, there were a couple of alternatives that could be used in Eclipse. The Eclipse Test & Performance Tools Platform (TPTP) is a widely documented profiling tool that is available as a plugin to Eclipse. This was tested, but several unsuccessful attempts at profiling led us to seek out alternatives.

Eventually JProfiler was chosen, although not available freely, a free trial period provided enough time to profile aspects of the ABM which were eliciting concern.

4.6.2 Version Control

From the start of the project, backing up software was treated as a priority. Although several standalone VCS clients were available, the eventual choice was Subclipse. Subclipse is another Eclipse plugin which integrates with the IDE and makes backing up code simple.

4.7 The ABM

As described in section 2.6.7 a RePast model consists of a Model, a Space and at least one Agent object. It was decided to make the space a spatial space, rather than an abstract space, and as such every agent has a location within this space defined by a pair of coordinates. Modelling the landscape spatially encapsulates the real world reasonably accurately - a grid is as much flexibility as RePast will allow - and giving households actual locations leaves scope for them being influenced by their neighbour's actions.

At its most basic, a household agent consists of seven attributes which describe it; its age class and values representing its heating system, dwelling type, construction type and whether or not it has window, wall and loft insulation. These are the house characteristics that need to be modelled by the stock transformation method.

4.7.1 Architecture of the ABM

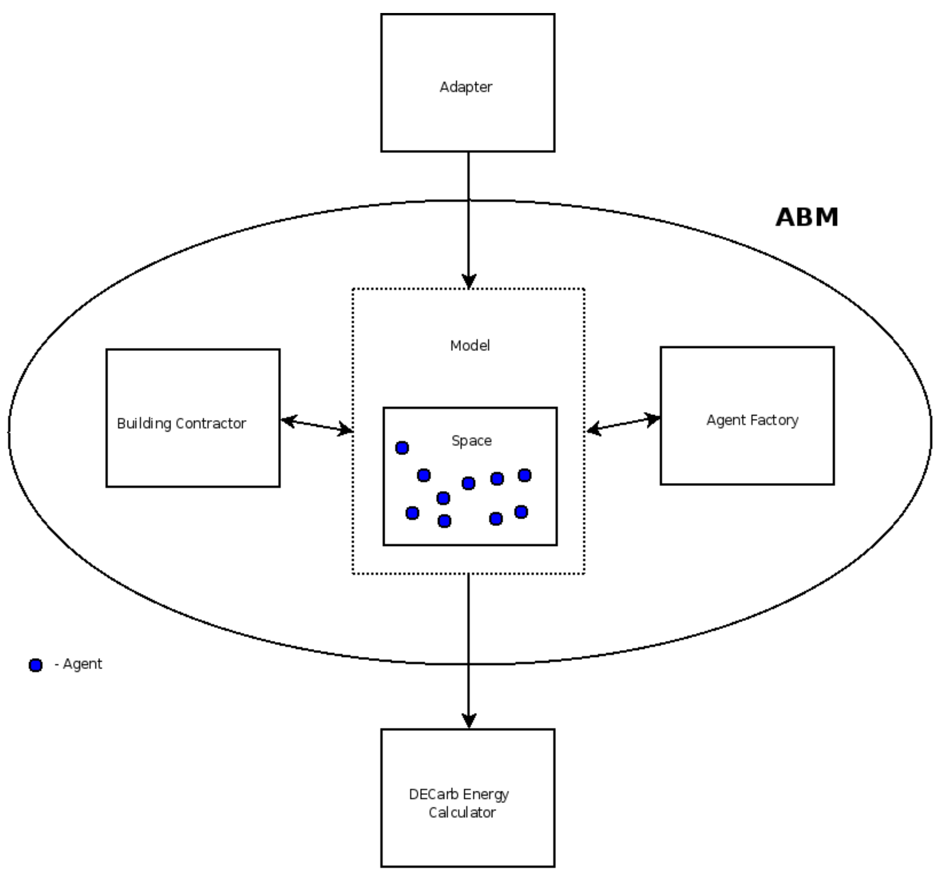

Figure 9: Architecture of the ABM

While it was felt important to keep the ABM as simple as possible, it was also important to avoid the temptation to put all of the code into the main model. As such, where it was possible, concepts were encapsulated within separate helper classes.

Agent Factory

Agent generation is encapsulated using the factory design pattern (Gamma et al., 1994). A static method in the factory returns an agent given only a base set of attributes and the age class that the agent should belong to. Implementing in this fashion ensured that agent creation was kept separate from the computation in the model, and as such any changes to the creation of the agents was entirely hidden from the model.

Building Contractor

This class deals with the task of demolition and building new properties. It is explained further in section 5.6.

4.7.2 Populating the Space

The make-up of each of the age classes for the 1996 housing stock is defined in separate .xls (Microsoft Excel) files and as such is static. For each age class, the number of households is not specified, only the proportion of the population that each type of household represents is recorded. Existing code in DECarb reads in these files and creates six objects of type ArrayList, each one specifying an age class. Knowing the structure in advance allowed the ABM to be designed to work with these as inputs.

The adapter passes each ArrayList to the model and using these figures, agents are generated in a **deterministic** order and each one is placed at a **non-deterministic** location on the grid.

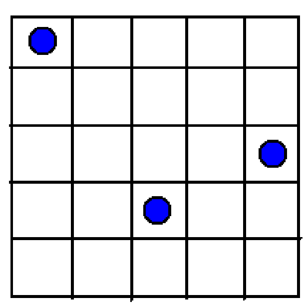

Figure 10: Agents Placed at Random Locations in the Grid

The effect of populating the grid in a deterministic order was not explored, and could potentially have caused the distribution to not be entirely random. In a situation where the location of an agent had a bearing on the output of the model and the results of the model were being used to prove a hypothesis, it would be necessary to test the effect of placing the agents in a deterministic order versus the effect of placing the agents in a non-deterministic order.

The EBM models four different regions, representing different areas of the UK. There are no dependencies between these regions, therefore they can all be modelled separately, although the model outputs should be aggregates from all regions. The ABM currently only contains one space, and so only models one region. To model all four it would be necessary to include four space objects within the model. This has not been done yet as it is seen as introducing too much complexity, for only a small gain in terms of accuracy and time.

Population Size

The original aim of this project was to model every individual household in the U.K. as an agent. Unfortunately due to hardware memory constraints, creating 23 million agents on one machine simply was not feasible. Therefore the ABM contains a parameter where the user can define how many households each agent will represent. So if the user choose a value of 200, every agent will represent 200 households and these will be treated as one entity. Any groups of households for which less than 200 exist are not modelled, although not ideal this constraint was enforced by hardware and unavoidable.

The model was created in a scalable manner, such that if it was run on a machine with the memory capability to cache 23 million agents it would still model the situation accurately. On a laptop with 1GB of RAM, around 3,000 agents could be created before an out of memory error is encountered.

3,000 agents is still a very small proportion of the starting population, and so additional savings needed to be made on memory. One of the advantages that it was felt would be gained by the use of JBoss Rules was that every agent would have its own working memory in computer memory. Unfortunately this was found to be a huge drain on the amount of free memory, and refactoring the code to place the decision-making rules in the Java code of the agent freed up a lot of memory. This implementation allowed for well over 20,000 agents to be created. Although the agent code was clearer when the rules were encapsulated within a JBoss Rules file, it did not offer any additional functionality, and the benefit of being able to model a larger number of households was greater than that of additionally clarity in the code.

4.7.3 How the Population is Scaled

The ABM is passed data specifying the size of the stock in 1996 and how this is split up between each age class. Ideally every household would be modelled as an agent, but due to the possibility of hardware constraints one of the model's parameters allows the user to specify how many households each agent represents. For this example take this parameter to be set to 200, and say that the size of the stock is 23 million dwellings.

The problem is then split up in to age classes, so for example if age class 1 is 10% of the total population, it represents 2.3 million dwellings. For each type of household a figure representing what fraction of the total age class this type of household represents is given (see section 4.7.2). For example, semi-detached households with no window insulation, no wall insulation, no loft insulation, and a gas heating system may represent 10% of age class 1.

$$2{,}300{,}000 * 0.1 = 230{,}000$$

$$\frac{230{,}000}{200} = 1150$$

Therefore 1150 agents would be created to represent 230,000 households. If, however, this type of dwelling only represented 0.00001 of the total dwellings in age class 1, the scenario would be slightly different.

$$2{,}300{,}000 * 0.00001 = 23$$

$$\frac{23}{200} < 1$$

In this case, a group of household of less than 200 will not be represented at all, as it is below the user-defined threshold. In the first case the simulation will start off with 1150 identical agents, and they will soon change their characteristics.

At the start of the simulation a scale factor is calculated, this is the figure which the number of agents should be multiplied by to obtain the number of households. It is not just 200, because it also takes into account households that did not get represented.

$$\mathit{scalefactor} \neq \frac{\text{households created}}{\text{agents created}} \quad \mathit{scalefactor} = \frac{\text{total households}}{\text{agents created}}$$

Total households refers to all households in the age class – even those which were not represented in the ABM (i.e. those for which less than 200 exist). This ensures that the correct number of total households is represented. At any point in the simulation, multiplying the agent population of an age class by the scale factor should provide the total number of households in that age class.

$$\mathit{time} = 0$$
$$\mathit{agents\,created\,for\,age\,class}\,1 * \mathit{scalefactor} = 2{,}300{,}000$$

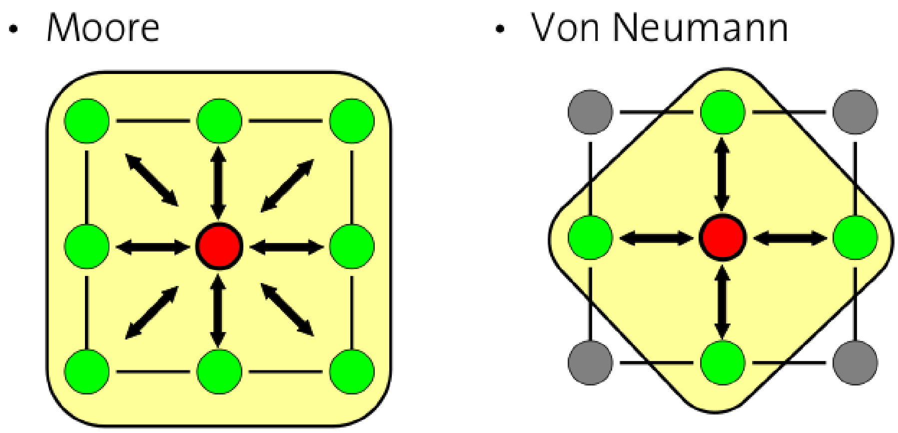

Figure 11: Moore and Von Neumann Neighbourhoods (Swiss Federal Institute of Technology, Zurich)

4.7.4 Awareness of Neighbouring Agents

When using a 2D space in RePast, there are library functions which return all of the agents within a certain distance of a grid point. When determining the neighbours of an agent, there a two different techniques, as demonstrated in figure 11. In this project any mention of neighbours should be read as Moore neighbours. A Moore neighbourhood was chosen over a Von Neumann neighbourhood to model a neighbourhood as this is the technique that appears to best encapsulate the real-world situation. There is no data to test this against and so this is simply an assumption made in the model.

A simple piece of face validation carried out to make sure that agents were aware of their neighbours was to make every agent print its neighbours at every step, which, if carried out on a small grid, is an easy process to validate.

4.7.5 Recording Summary Data

In order for the existing DECarb energy calculator to calculate energy consumption and carbon emissions it requires a summary of the housing stock data for each age class that exists at each period. The data is measured at the years 1996, where only 6 age classes will exist, 2000, where 7 will exist, and then every 10 years until 2050, where 12 age classes will exist. The model calculates and stores the state of the housing stock at each of these points in the format which is expected by the DECarb back-end.

For this reason, no adapter class is required to sit in-between the ABM and the DECarb energy calculator, the data is already in the correct format.

The DECarb energy calculator is discussed in more detail in Natarajan & Levermore (2007a, pp. 2). It is based on the BREDEM model, but with some newer algorithms to compensate for the fact that many of the BREDEM algorithms are based on older data. As with the DECarb GUI, this component is treated as a black box in this project - age class data is fed in, and data listing carbon emissions and energy consumption is output.

4.7.6 The Concept of Time in a Simulation

In an ABM, the concept of time is replaced by a sequential event known as a step. A step will not start until the previous step, and all of its associated tasks, has finished. At each step events can take place in the model and for each agent. Events can also be scheduled to take place every n^{th} step.

In this simulation each step represents 1 year, so the length of one run of the model is 54 steps, 1996-2050. The only other feasible option was for 1 step to represent 1 month and this seemed to be an unnecessarily large number of steps. The fact that the DECarb GUI allows the user to define values at intervals of 10 years, further suggests that data is not available at a granular enough level to support monthly steps.

4.8 Conclusion

The aims of this cycle were met by creating an ABM of passive agents, fully integrated with the existing DECarb code. Having this basic system in place allowed for further components of the model to be added iteratively in further cycles.

Chapter 5

Cycle 2 – Implementing Marionette Agents

The aim of the second cycle was to make the agents adhere to the user specified distributions for each of the house characteristics. Once a suitable technique was chosen and implemented, a stock transformation method, hopefully capable of docking with DECarb, would have been created.

5.1 Requirements For the Second Cycle

1. Achieve the user specified distribution for one of the four main housing characteristics (i.e. window insulation)
2. After this is done, implement this for the other 3 major values of the base set (Heating System, Wall Insulation, Loft Insulation)
3. Implement a demolition model that demolishes the oldest properties at a user specified rate

Achieving the user specified distribution was one of the most important aims of the whole project, as the very idea of a stock transformation method is to accurately model how these attributes change over time. The rationale behind this set of requirements is that when this first requirement was complete, extending the method to apply to other house characteristics would be trivial in terms of complexity - although possibly time consuming. The final aspect requiring consideration from the EBM stock transformation method was that of demolition, so a framework had to be design and implemented to deal with that.

At this point it is important to recall the definition made between new (those built since 1996) and existing (those built pre-1996) dwellings. When modelling a household, there are six observables, these are listed in table 5. Two of these, the user may only define for new dwellings, they are dwelling type and construction type. For example, the user can define the fraction of new dwellings that will be of dwelling type 1, what fraction will be of dwelling type 2 and so on. The rationale behind this is that existing dwellings will not ever change their state, i.e. a semi-detached house will very rarely become re-classified as a Bungalow.

There are a further four observables for which the user can define uptake for new and existing dwellings, these are wall insulation, window insulation, loft insulation and heating system. For these observables the principle for new dwellings is the same, but there is now an additional malleable graph, by which the user may define how the population of existing dwellings may alter its uptake of these over time. For example the user may specify that 40% of households have single-glazing in 2020 and this will fall to 20% in 2030. So the challenge in creating an accurate stock transformation method is to find a technique by which half of 2020's non-double-glazed households will adopt double-glazing by 2030.

Table 5: Observable values to be modelled

Observable	Type of observable	Possible Values	Can user define new, existing or both?
Window Insulation	Binary	Single-glazed, Double-glazed	Both
Loft Insulation	Binary	Insulated, Uninsulated	Both
Wall Insulation	Binary	Insulated, Uninsulated	Both
Heating System	6-Fold	Gas, Electric, Community, Heat Pump, dCHP (Stirling), dCHP (Fuel Cell)	Both
Construction Type	6-Fold	Solid Masonry, Cavity Masonry, Concrete, Timber, Metal, Other	New
Dwelling Type	7-Fold	End Terrace, Mid Terrace, Semi-Detached, Temp, Purpose Built Flat, Converted Flat	New

5.2 Achieving the Distribution of Agents

At this point in the project, an important decision as to type of agents to implement had to be made. As previously discussed, an ABM is traditionally used for bottom-up modelling, with agents only having local information available to them, essentially modelling the real world as closely as possible. But, in order to achieve the objective of meeting the user specified distribution for house characteristics, a traditional approach did not seem to be feasible.

Modelling the heterogeneous micro-behaviours of agents is an effective method of discovering emergent behaviour in a society, but that is not the problem being explored. Given a scenario of how energy-related technologies will be adopted, the goal is to see which proportion of the population adopt these, not how many of the populations adopt these. For example, at this stage of the project, the emergent property of interest is not the fact that 70% of the population will obtain window insulation, but *which* 70% of the population obtains window insulation.

As such, the modelling done was not carried out with what Gulyas classes as bounded rational agents, but actually with what he specifies as marionette agents.

"agents without autonomous control...behavior is guided by system level aggregate properties." Gulyas (2006)

Using marionettes allows for a re-creation of what is being done in the EBM. At a global level, every agent can be assigned a probability, defining the chance with which it will adopt each insulation at every step. Parunak et al. (1998) promote this technique as a middle-ground between ABM and EBM when talking about how the behavioural decisions of an agent can be determined by evaluating equations. Agents will have no local knowledge, or global knowledge for that matter, they are just marionettes acting as they have been instructed to. It is important to stress that this is not the end goal of the project. Using marionettes was a means to the end of building confidence in the abilities of the ABM stock transformation method, and paved the way for further exploration of using an ABM for stock transformation.

5.3 Getting the Distribution of Agents Correct – Method 1

When choosing a method by which to get all of these distributions correct, it was thought best to start with a simple binary attribute, and move on to the more complex attributes when a suitable technique had been found. Window insulation was chosen to experiment with, and a suitable equation-based decision making system needed to be devised.

Given a period of length N, a starting level of single-glazed houses Y_1 and a desired ending level of single-glazed houses Y_N, P_N represents the fraction of Y_1 that will be represented in Y_N.

$$P_N = \frac{Y_N}{Y_1}$$

This represents which fraction of agents should get double-glazing over N years, so to work out what fraction of agents get double-glazing per year, we must divide by N, as shown below.

$$U = \frac{P_N}{N}$$

This gives U, which is the fraction of agents that must adopt double-glazing per year. This is demonstrated in the example below:

$$N=2$$
$$Y_0=0.4$$
$$Y_2=0.2$$
$$Y_N=Y_2$$

$$P_2=\frac{0.2}{0.4}=0.5$$
$$U=\frac{0.5}{2}=0.25$$
$$Y_0\cdot(N\cdot U)=Y_2$$

If the fraction of the population with single-glazing in year 0 is 0.4 and should be 0.2 by year 2, then 0.25 of those with single glazing need to adopt every year.

5.3.1 Results

To test this method, decision-making logic as demonstrated below was carried out by every agent on every step.

```
if(notGotWindowInsulation && (randomNumber <= U))
{
   getWindowInsulation
}
```

This test was carried out with 100 households over 30 runs and the results can be seen in figure 12. As it is clear to see, using this method produces a curve of roughly the right shape, but the generated values are consistently too low when graphed against the expected values.

This was due to the fact that the decision making logic ensures that U of the single-glazed population gets double-glazing every year. But the size of the single-glazed population decreases every year, resulting in not enough of the population obtaining double-glazing.

5.4 Amortization

The method chosen instead is the solution to the well known compound interest problem. The standard compound interest formula takes a present value PV, a number of periods n and

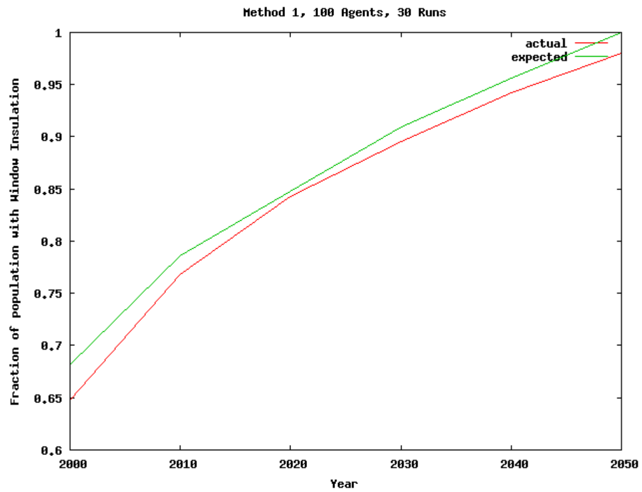

Figure 12: Stock Transformation Method 1 with 100 agents over 30 runs

an interest rate I as knowns and returns a future value FV (see Appendix B). A brief re-arrangement of this formula presents us with a new formula in I, as shown in figure 13.

This formula calculates the compound interest rate I achieved if an initial investment of PV returns a value of FV after n accrual periods.

$$I = \sqrt[n]{\left(\frac{FV}{PV}\right)} - 1$$

Figure 13: Formula for Amortization

The value I may now be used on every round, for any Y_i.

$$Y_i = Y_{i-1} + (Y_{i-1} \cdot I)$$

So to use the previous example:

$$N=2$$
$$Y_0=0.4$$
$$Y_2=0.2$$

$$I = \sqrt[2]{\frac{0.2}{0.4}} - 1 = -0.292$$
$$Y_1=0.4 + (0.4 \cdot -0.292) = 0.2832$$
$$Y_2=0.2832 + (0.2832 \cdot -0.292) = 0.2$$

In this example the fact that the set of single-glazed households decreases over time is compensated for, so rather than the previously calculated 0.25, using amortization 0.292 households move per year.

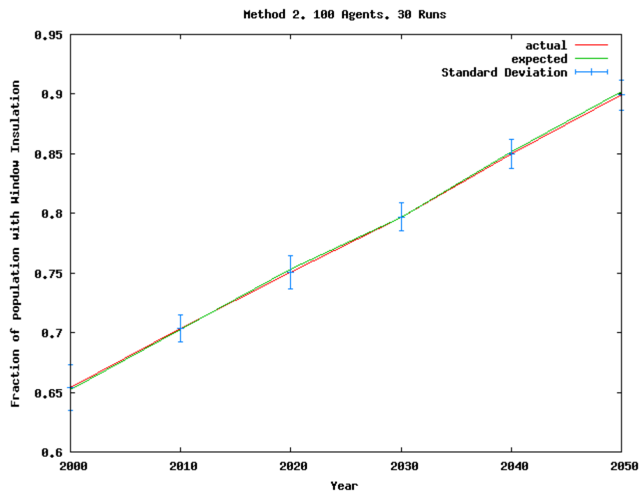

Figure 14: Method 2 with 100 agents over 30 runs

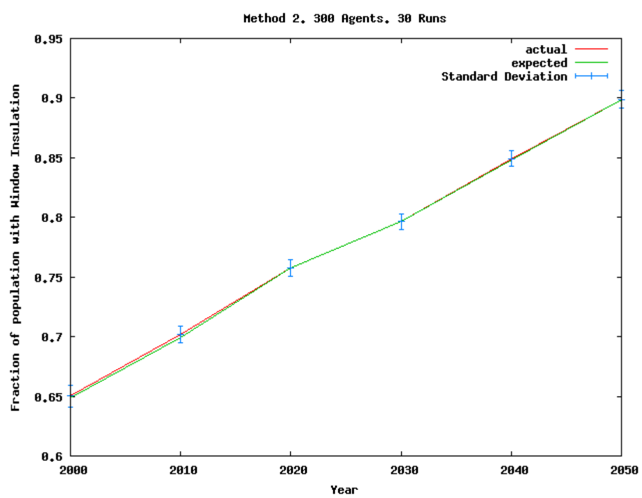

Figure 15: Method 2 with 300 agents over 30 runs

Results

With this new method to compute the the number of houses to be fitted with double-glazing per period, the original tests were re-run. The results, as seen in figure 14, were exactly as had been hoped in terms of their close correspondence to the expected results. When being run with 100 agents the standard deviation was too high to be acceptable. But the test was re-run with 300 agents, and as shown in figure 15, the standard deviation decreased dramatically, and with the fact in mind that the final simulation would be run with many more than 300 agents, this was not considered an issue.

5.5 Extending Amortization to Attributes with Multiple States

Although at a glance, extending the amortization formulations to use on attributes with multiple states may appear a trivial task, the method used is considerably more complex.

There are six different heating systems modelled. A household can be in any of those six states, and at every period in time where they can re-consider their heating system, there is a different probability of changing to each of the other states. This creates quite a complex scenario because not only is there a different probability of going to each state, but these probabilities change depending on which state the dwelling is currently in.

To demonstrate this, the simplest example to use is an attribute with four states as seen in table 6. The number of households in states A and B decreases over the time period, while the number of households in states C and D increases over the time period.

Table 6: Distribution of the population over attribute with 4 states

State	2000	2010	Change
A	0.25	0.1	-0.15
B	0.25	0.2	-0.05
C	0.25	0.3	+0.05
D	0.25	0.4	+0.15

Table 7 shows the probability of an household moving state over the time period from 2000 – 2010. There is a different set of probabilities for households in each state and this means that during a model step, a household's current state must be checked before it is assigned a probability of changing states. The figures in table 7 must be further manipulated to take into account amortization, as the numbers specified are over the whole period of 10 years.

Table 7: Probability of any household changing state over the time period

State	A	B	C	D
Currently A	0.4	0	0.2	0.4
Currently B	0	0.8	0.1666	0.3333
Currently C	0	0	1.0	0
Currently D	0	0	0	1.0

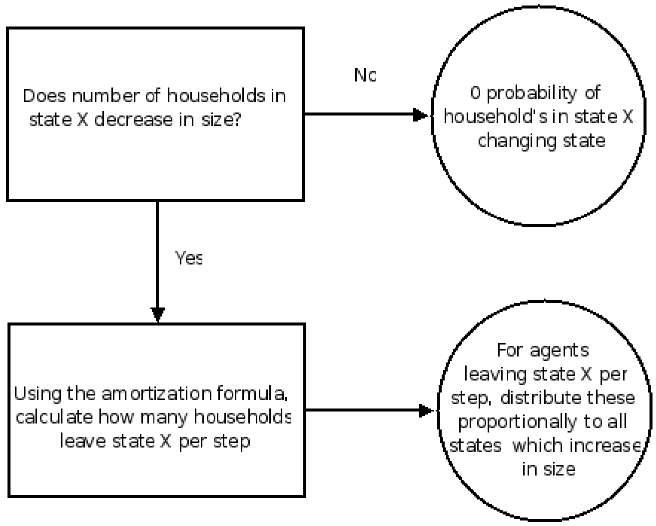

Figure 16: Algorithm for dealing with attributes with multiple characteristics

5.5.1 Algorithm

The decision making process for calculating the probability of an agent leaving each state is shown in figure 16. It is based on the simple formula devised for amortization, but then extended further to account for more complex multi-attribute situations.

If the number of households in a state is going to increase in size, the situation is very simple, the probability of any of the households in it changing state is 0. If the number of households in a state decreases in size, the first step is straightforward. Calculating what proportion of the state will leave at each step uses our amortization formula, the last step however, is not as simple as it sounds.

5.5.2 Implementation

The complexity in this algorithm comes about because before the task of '*distribute proportionally to all states which increase in size'* can be carried out, a full loop of the states must first be done to ascertain the total number of households moving **to any state**. Once the total fraction of households moving to any state is known, it is a simple task to distribute proportionally amongst them, as demonstrated with the pseduo-code below:

```
while ( state < numberOfStates)
{
  if(state decreases in size)
  {
    move[state] = agents leaving per step
  }
  else
  {
    agentsMoving += amount state increases in size
    move[state] = 0
  }
  next state
}

while ( state < numberOfStates)
{
  if(move[state] == 0)
  {
    return stayInSameState
  }
  else
  {
    while ( state2 < numberOfStates)
    {
      proportionOfTotal = increasesInSize(state2) / agentsMoving
      proportionOfStateLeaving = agents leaving per step *
        proportionOfTotal

      returnArray[state2] = proportionOfStateLeaving
      next state2
    }
    return returnArray
  }
  next state
}
```

5.5.3 Results

Using this algorithm to update the probability of an agent changing state for multi-state attributes provided very accurate results as seen in figure 17. Graphs showing the tracing of more than 3 attributes can be seen in appendix D, figures 42 & 43.

The techniques detailed in this section provide an effective way to track a user-specified distribution of any household characteristic over the length of the simulation. Implementing

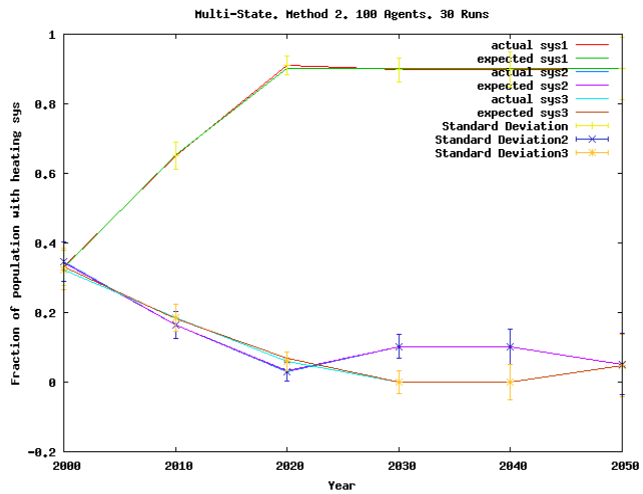

Figure 17: Result Graph Using Multi-Attributes. 100 Agents. 30 Runs

these in the population of marionette agents should ensure that the population adheres to the user-specified distribution for all characteristics.

5.6 Dealing with Demolition

In order to effectively dock with DECarb's EBM stock transformation method, considering demolition was very important. Demolishing an old, inefficient dwelling and replacing it with a new, better-insulated dwelling can, if done many times, make a noticeable change to global carbon emissions.

In the existing EBM, demolition rates are calculated very simply. The user defines three different variables; the total population of the UK, residents per dwelling and demolition rate per annum. From these 3 input sources, DECarb calculates the total population per year, and how much of this consists of new dwellings and how much consists of existing dwellings. This is treated as a black-box process and the numbers are fed straight in to the ABM.

Once the demolition rates have been calculated in the EBM, the actual process of modelling demolition is very simple. Dwellings are demolished at random from the oldest age class (age class 1). Once all of the dwellings in this age class have been exhausted, dwellings from the second-oldest age class are demolished and so on. It was felt that this is an overly-simplified scenario, and could possibly be modelled more accurately. Although in this section, the aim is to merely replicate the behaviour of the EBM, in chapter 7 the possibility of using the ABM to model different demolition scenarios is explored.

5.6.1 The Building Contractor Class

The task of demolishing dwellings and building new dwellings is encapsulated in the class BuildingContractor. This class was designed with the fact in mind that different demolition scenarios may be explored later in the project's life cycle. For that reason, it takes as parameters either a list of agents to be demolished or a number of new agents to be built. The task of choosing which agents are to demolished is carried out in the model, allowing BuildingContractor to be used in exactly the same way in every scenario. The BuildingContractor deals with two main tasks; creating new dwellings and demolishing dwellings.

Creating New Dwellings

Creating new dwellings is the simpler of the two tasks that are carried out by BuildingContractor. Passed an empty list and a number of new dwellings to create, it has the task of populating that list with new dwellings. To be recognised as new, a dwelling must have its age class set to the period in which it is constructed. An additional facet to this situation is that the user may define what fraction of new dwellings has the which window insulation, loft insulation, wall insulation, heating system and what type of dwelling it is and what it is constructed from.

Receiving figures specifying that 60% of new dwellings should have double-glazing and 40% should have single-glazing, a random double between 0 and 1 is chosen. If the number is greater than 0.6 then the new dwelling is attributed with single-glazing, and if not; double-glazing.

The alternative to this technique would be to keep a count of how many of each type of dwelling need to be created, i.e. 6 dwellings to have double-glazing, 4 to have single-glazing, and decrement these when one was created. The reason that this method was not chosen is that it would impose additional storage and calculation overheads on a large population of agents and there is a chance it could have resulted in a determinism in the demolition process.

Demolishing Dwellings

In order to cut down on memory and garbage collection overheads the BuildingContractor does not actually remove any agents from the model. All of the agents passed to the BuildingContractor have their attributes altered so that they appear to be new dwellings. This involves the model having to alter the number of new dwellings that it tells the BuildingContractor require building, as seen below.

```
DwellingsToCreate = X
DwellingsToDemolish = Y

BuildingContractor.demolish(Y)
BuildingContractor.newBuild(X - Y)
```

A problem can occur in this situation if the number of properties to be demolished is greater than the number of properties to be created. The BuildingContractor clearly cannot be told to create a negative number of dwellings. In this scenario there is an additional parameter in the demolish() method which allows the model to specify how many of the dwellings should be replaced by new builds.

Once the building contractor has the dwellings which need to be converted to new dwellings, it uses the same technique as specified for creating new dwellings to calculate which attributes to change.

5.7 Conclusion

The aim of this cycle was to carry out the design and implementation of a similar stock transformation to the existing EBM, using ABM. With the discovery of amortization as a method for agent decision making, and the replication of the existing demolition scenario, this aim has been met.

Amortization has shown to be accurate with only a small standard deviation and the integration of all of the functionality shown in this cycle should culminate in a convincing set of results when the ABM is docked with the EBM.

Chapter 6

Back-Casting

After the completion of the first two cycles, the first goal of the project had been achieved. In order to generate full confidence in the model, and meet the second goal, it was important to carry out validation in order to ensure implementation was carried out correctly.

6.1 Requirements for the Back-Cast

1. Carry out face validation (see section 2.5.1) on the ABM
2. Dock (see section 2.5.2) the output generated by DECarb using the ABM module with the output generated by DECarb using the EBM module in a back-cast scenario from 1996-1970
3. Validate the ABM module by back-casting (see section 2.5.2) against historical data from 1996 – 1970.

Face validation was necessary early-on just to look out for obviously erroneous data. Once the data looked intuitively correct, the aim of docking with the existing EBM-based system was to see whether the results were in the correct range. The aim of validating against recorded historical data was to give an exact idea of the accuracy of the ABM-based system. The end goal of this phase of the project was to create confidence in the accuracy of the ABM module, therefore lending more credibility to results calculated when using it for forecasting.

6.2 Back-Casting

In order to create confidence in the model's forecasting ability it is first necessary to look back. By starting in 1996 and working backwards it is possible to 'predict' the energy consumption and carbon emissions in 1970. Of course this is not a prediction as such, but it is a scenario for which concrete data exists to test against. This technique is called back-casting and is essentially the opposite of a forecast. Back-casting is explained in section 2.5.2 and Natarajan & Levermore (2007a, pp. 6) give a full description and justification of back-casting as a suitable validation technique in this scenario.

6.2.1 Considerations When Back-Casting

Although the process of back-casting is very similar to forecasting, there are a few additional factors to take into account when back-casting. First of all, no new dwellings should be created. Secondly, when demolishing properties, those in the newest age class (age class 6) rather than the oldest age class (age class 1) will be demolished. In this scenario, the process of 'demolition' is not modelling the demolition of properties, but rather accounting for the fact that properties built between 1970 and 1996 should not exist at the end of the back-cast.

Apart from these factors, the model should behave in exactly the same way. This is essential, because if the computation carried out for back-casting was entirely different to that for forecasting, a successful back-cast could not, justifiably, create confidence in the forecast.

6.2.2 Specifying the Back-Cast Scenario

To accurately model the transition from 1996-1970, the exact scenario that occurred between 1996 and 1970 is specified via the DECarb GUI. For instance it is known exactly how the size of the population changed in that time and data exists on the uptake figures of each of the technologies in this period.

This data was all obtained by Natarajan & Levermore (2007a) and simply loaded into the DECarb GUI using a configuration file. This ensured that back-casting was carried out with the exact same data as used by DECarb.

6.3 Docking with DECarb

"If docking an unproven model with a valid model, an agreement between the two models can infer the validity of the unproven model" (Xiang, 2005)

A successful docking process with DECarb, would, as Xiang states, infer the validity of the

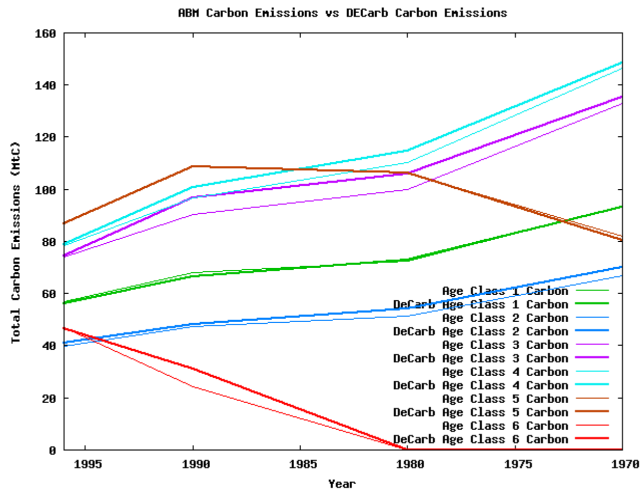

Figure 18: Back-Cast of Carbon Emissions for each Age Class

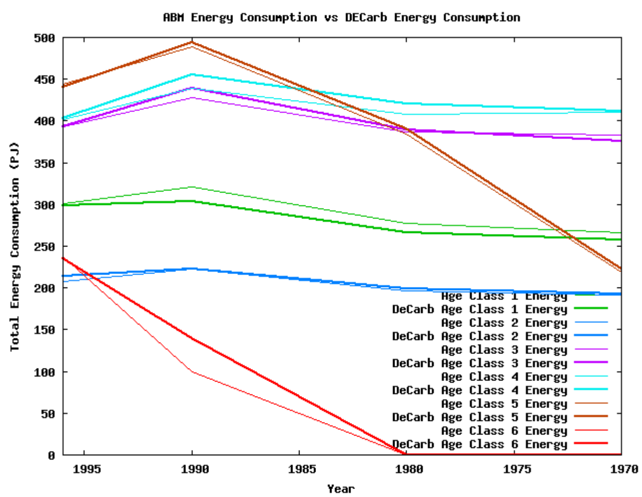

Figure 19: Back-Cast of Energy Consumption for each Age Class

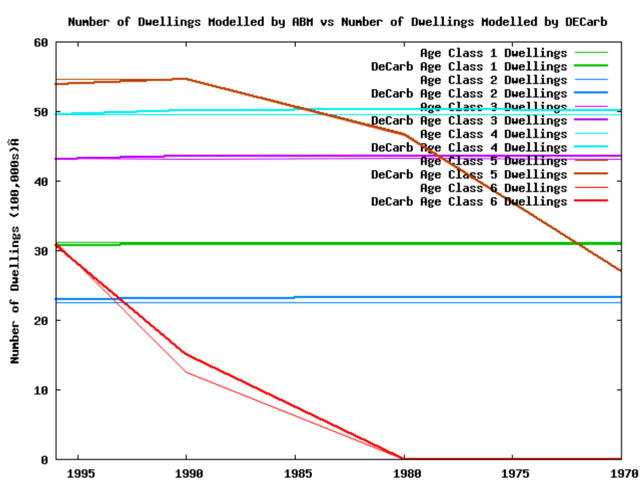

Figure 20: Total Dwellings Modelled by ABM vs DECarb

unproven ABM. Although at the highest level, docking just consists of comparing the results of the two simulations, the values comprising the results of the simulations will also be docked. Before looking at the macro details of the nation, it will be verified that each age class of the population has been modelled correctly. This adds an additional layer of transparency as to how the results were obtained and adds further credibility to the ability of the ABM stock transformation method.

The results obtained in this section are the mean averages of twenty runs of the ABM. The DECarb energy consumption figures are from a single run because, being deterministic, DECarb will always produce the same results from the same inputs. All historical data is taken from Natarajan & Levermore (2007a). For energy consumption, the actual figures are used, but for carbon emissions not actual figures were recorded. The figures used instead are from the Domestic Energy Fact File (DEFF).

Using the ABM stock transformation method produced results very similar to those of DECarb when looking at carbon emissions for every age class, as seen in figure 18. The figures calculated for age classes 2 and 3 are slightly lower than those predicted by DECarb, and it is not clear why, as the number of dwellings modelled was very similar (see figure 20). The deviation seen between results is not necessarily a negative factor, as DECarb is just another model of the same scenario and, as such, minor differences are to be expected.

As demonstrated in figure 19, the comparison between DECarb's predicted energy consumption and the predicted energy consumption of the ABM provides similar results. The differences between predictions, particularly for age classes 1 and 6, are greater than with the carbon emissions, and there is no obvious explanation as to why this is. The overall trend suggests that the ABM stock transformation model behaves in a similar enough manner to DECarb's existing EBM stock transformation model to make its forecasting ability credible.

6.4 Comparison against Historical Data

As seen in figure 21, the results provided by using the agent-based stock transformation method were hugely promising. Interestingly, the energy consumption calculated on the ABM in 1996 is marginally lower than that calculated with the EBM, even though they both use the same base set of data – which should have been untouched in 1996. This could be due to the fact that not every single household is modelled individually, and those with a very small representation are not modelled at all. Both EBM- and ABM-based calculations vastly over-estimate the actual energy consumption in 1990 and Natarajan (2007a, pp. 8) puts this down to a warmer than usual summer which resulted in relatively low energy consumption. Although twenty runs is a relatively low number, the small error bars are evidence enough that these results are an accurate representation of the capabilities of the

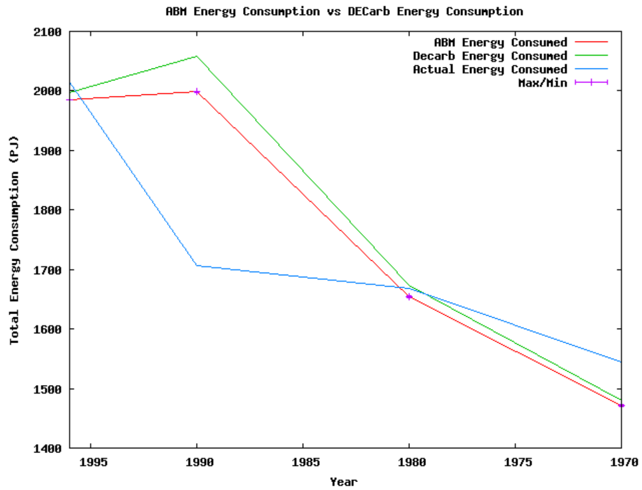

Figure 21: Energy consumption with ABM transform in back-cast scenario

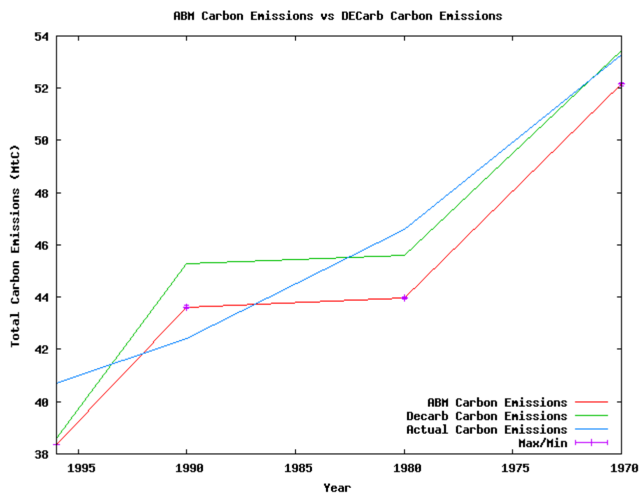

Figure 22: Carbon emissions with ABM transform in back-cast scenario

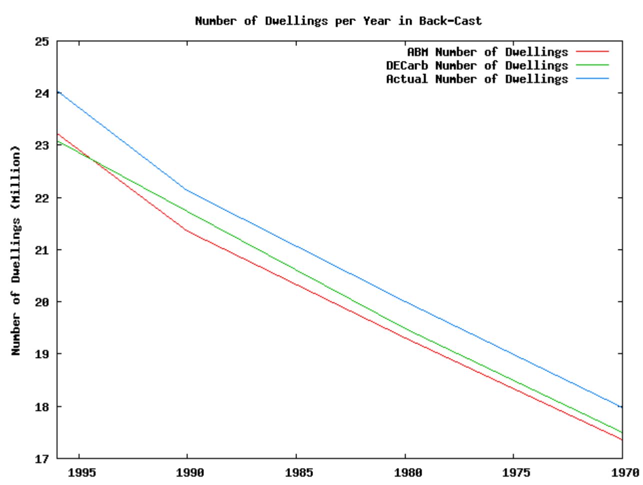

Figure 23: Number of Dwellings Modelled in Back-Cast

ABM stock transformation method. The results in figure 22 demonstrate that using the ABM stock transformation method also provides promising results when calculating carbon emissions, although, as when predicting energy consumption, it appears to under-predict when compared to the actual figures. As with energy consumption, in 1990 the DEFF carbon emissions are lower than predicted, and Natarajan attributes this to the same factor.

In both figures 21 and 22, the numbers calculated are – generally – marginally lower than the actual figures. This could be caused by the fact that both EBM and ABM model a slightly lower number of dwellings than actually existed, as demonstrated in figure 23. However, this is unlikely, as both models only exclude dwellings who's representation is too small to make a reasonable difference. As previously stated, in the case of the ABM this is a requirement derived from hardware constraints.

6.4.1 Deviation From Actual Results

Using figures obtained from Natarajan & Levermore (2007a), it is possible to calculate the deviation of the ABM results from the actual figures from this period (a break down of figures can be seen in Appendix D, tables 13 & 14).

Table 8: Averages of deviation shown by EBM and ABM

	Carbon Emissions Deviation from DEFF	Energy Cons Deviation from actual
ABM	-2.65%	2.51%
EBM	-0.9%	-5.4%

As shown in table 8, the results calculated using the ABM stock transformation method are more consistent than those obtained using the EBM stock transformation method. The total deviation from the actual results is 2.51% for energy consumption, and the carbon emissions prediction is -2.65% away from the DEFF figure calculated for the period 1970-1996. Seeing as the numbers calculated for 1990 are clearly erroneous, it is worth while adjusting the average figures in order to see what difference removing these numbers make.

Table 9: Averages of deviation shown by EBM and ABM without 1990 data

	Carbon Emissions Deviation from DEFF	Energy Cons Deviation from actual
Adjusted ABM	-4.47%	-2.36%
Adjusted EBM	2%	-1.3%

Adjusting the figures to remove the data from 1990 gives a more accurate representation of the situation, as demonstrated in table 9. Both of the ABM averages are under-predictions, which mirrors the general trend seen in the figures.

6.5 Conclusion

The successful docking with DECarb has demonstrated that an ABM stock transformation method can be used as an effective replacement for the existing EBM stock transformation method. As well as providing very similar results to the EBM, there are several other advantages to using the ABM. Primarily, the ABM has a theoretical grounding, in that every household is being represented as a Java object, rather than just a number in a matrix.

Additional advantages lie in the flexibility and scope for heterogeneity that an ABM offers over an EBM. Now that a basic ABM has been established, the scope for what can be explored is limitless. The following sections demonstrate both cited benefits, as first the ABM's flexibility is used to explore different demolition scenarios and then households are assigned with heterogeneous behaviours in order to model how they will react to changes in their environment.

Chapter 7

Cycle 3 – Exploring Demolition Scenarios

After completing the first two cycles, Natarajan's two concrete aims were met. As each of the additional goals were equally weighted in terms of priority, some amount of flexibility was afforded as to what to explore next. After an evaluation of the relative merits of each goal, as seen in section 3.4.4, the exploration of demolition scenarios was chosen as the first to look at.

7.1 Requirements For the Third Cycle

1. Create a loosely coupled demolition sub-model which allows different demolition scenarios to be explored
    1.1. Model should allow oldest household's to be demolished
    1.2. Model should allow for random selection of household's to be demolished
2. Develop a metric to classify how well insulated a household is
3. Integrate this with the sub-model to allow the 'worst' set of households to be demolished
4. Dock demolition sub-model outputs with 40% House outputs

7.2 Further Exploration of Demolition

In the existing EBM the method that decides which dwellings are to be demolished is very simple. A number is generated as to how many dwellings must be demolished, and this number of dwellings is removed from the oldest age class. If the oldest age class is empty, dwellings are removed from the second oldest age class and so on. This is clearly a gross oversimplification of the scenario, in reality there are many factors which would dictate which dwellings get demolished at any point in time.

Bender (1979) says that a housing producer will demolish an existing dwelling as soon as the difference between its net present value, and the net value of a new dwelling, becomes greater than the cost of demolition. This calculation requires data streams that are not available in the existing simulation, but it is feasible to explore what effects demolishing housing with lower quality insulation will have on calculations.

The flexibility afforded by using an ABM stock transformation method allows us to explore alternative demolition scenarios, with very little effort from a coding point of view.

7.3 Demolition Scenarios

Demolition Scenario 1 - Oldest Properties

One demolition scenario was the existing model of demolition, in which the oldest properties are chosen. Whilst, as earlier stated, being an oversimplification, this technique has some logical grounding. An older property is more likely to be in a poor state of repair, and so less appealing to potential purchasers. This loosely ties in with Bender's findings, as if a property is less appealing, its net present value is likely to be low. It is still far too general to be regarded as an accurate model, as there will likely be houses built in other age classes with just as poor energy saving technologies.

Demolition Scenario 2 - Random Properties

A very easy case to test is that of demolishing random properties. Of course, there is no theoretical basis for this, and it does not accurately encapsulate the real world situation. The benefit of modelling this scenario is that it can show how much difference there is between demolition scenarios. If it turned out that the random case was exactly the same as the demolishing the oldest dwellings, it would suggest that creating an accurate demolition sub-model was not an important aspect of modelling the housing stock, and this could be taken into account in any further work.

Demolition Scenario 3 - 'Worst' Properties

An interesting third scenario to look at was demolishing the most poorly insulated properties. This scenario first required the development of a metric to determine which households were most poorly insulated. Out of the three proposed scenarios, this one most closely models the real situation, although creating a truly accurate metric is well beyond the scope of this project.

When developing the metric it was decided to bound it to no more than three dwelling characteristics. Although heating system, dwelling type or construction type could have been used, the fact that these are multi-value attributes counted heavily against them. Firstly there was no clear 'good' or 'bad' state to be in, for example, is a property with a dHCP (Stirling) heating system more desirable than a property with a dHCP (Fuel Cell) heating system? Secondly, the fact that each has many values means that even if a ranking for each was created, the metric would become very complex with 6*6*7 = 252 possible combinations to consider.

As such, the three characteristics chosen were; window, wall and roof insulation. In each of these cases, having insulation is clearly preferable to having no insulation, and the fact that each of these cases are binary means that only 8 combinations needed to be taken into account.

In order to keep the process simple, each dwelling was categorized into one of four classes of demolition suitability. Dwellings with a demolition suitability of 1 are demolished first, and then those with a demolition suitability of 2 and so on. The allocation process is described below:

```
if ( dwelling has 0 out of 3 insulations )
  demolition suitability = 1
else if ( dwelling has 1 out of 3 insulations )
  demolition suitability = 2
else if ( dwelling has 2 out of 3 insulations )
  demolition suitability = 3
else if ( dwelling has 3 out of 3 insulations )
  demolition suitability = 4
```

7.4 Implementation

The ABM was designed with this potential work in mind (see section 5.6) and as such the amount of code that needed to be changed to include these scenarios was minimal. Demolition scenario was added as a parameter, and one method in the model had to be altered to take this parameter into account. This method, getAgentsToBeDemolished(), returns a list of agents to be demolished, which is then passed to the BuildingContractor.

Using this design means that the BuildingContractor class needs no knowledge of the demolition scenario at all.

The model contains references to all existing agents at the current point in time. An instance variable of type ArrayList, contains an ArrayList for each age class, and within each of these age class lists are the existing agents for that age class. Obtaining the list of agents to be demolished for demolition scenarios 1 and 2 is very simple, and does not warrant discussion, but getting the list of agents to be demolished for demolition scenario 3 is more complex.

Demolition Scenario 3

A naïve implementation may simply iterate through each age class list of agents and find those with lowest demolition suitability. But this technique would incur huge computation overheads as the size of these lists can easily run into the thousands. Another alternative would be to store a list of all agents with demolition suitability 1, and just take agents to be demolished from this list. This technique has additional, unnecessary, storage overheads, as well as computation overheads to update it every time demolition takes place.

A technique was chosen which utilised the model's existing system for storing agents, an ArrayList containing an ArrayList representing each age class. Before this stage, these lists were unordered, and so their ordering would not effect any other facets of the model. It was decided to place all agents with a demolition suitability of 1 at the start of each of the lists, followed by all of the agents with demolition suitability 2 and so on. This way it would be a simple case of removing an agent from the start of the list when one was needed for demolition.

Rather than spending time implementing a sorting algorithm, the Agent class was instead altered to implement the Comparator interface. This forces the implementation of a comparison method which imposes a *total ordering* on some collection of objects (Java API).

```
public int compareTo(Object otherAgent)
{
  if (getDemoSuitability() == otherAgent.getDemoSuitability())
  {
    return 0;
  }
  else if (getDemoSuitability() > otherAgent.getDemoSuitability())
  {
    return 1;
  }
  else
  {
    return -1;
```

```
  }
}
```

A list full of comparable objects can be sorted using the Collections.sort() method. Of course this sort does still incur computational overheads, but it uses a modified merge sort algorithm and can guarantee $n\log(n)$ performance (Java API). In order to make sure that this method is not called unnecessarily, the call to Collections.sort() is only made in the method that retrieves agents for demolition scenario 3.

In order to choose an agent, a non-empty age class is chosen at random to take the agent from. The other non-trivial task in this problem involves checking how many of each classification of demolition suitability exist in each age class, so that it is known instantly if any agents of the correct demolition suitability will be found in the age class list.

At each significant period (e.g. 1996, 2000, 2010, ... , 2050) every agent is iterated through in order to obtain the data that is needed by the DECarb energy calculator (see section 4.7.5), as the agents are already being iterated through at this point, there is very little additional overhead in counting how many of each class of demolition suitability exist in each age class. Storing these numbers quickens the process of selecting the agents to be demolished, as demonstrated below:

```
ageClass = (0 <= random < number of age classes)
currentLevelOfDemolitionSuitability = 1

if( ageClass has agents with currentLevelOfDemolitionSuitability)
{
  index = (0 <= random  < number of agents with demo suitability)
  use agent at index
}
else
{
  if( we have not tried all age classes)
  {
    start again from top with different age class
  }
  else
  {
    increment currentLevelOfDemolitionSuitability
    start again from top
  }
}
```

Using this technique returns a list of agents with the ideal demolition suitability, in a short amount with time with minimal computational overheads.

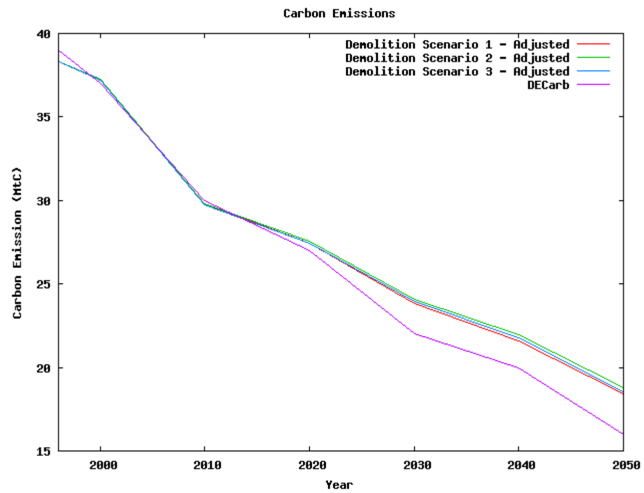

Figure 24: Adjusted Carbon Emissions

7.5 Results

Testing the validity of the different demolition scenarios using back-casting was not feasible, because demolition in a back-cast just consists of demolishing the newest dwellings. This is in order to realistically model the fact that houses built in the newest age class would not be present in 1970.

Instead, the scenario defined in the 40% house project (see section 2.2) was used. The results were then docked with those of the UK Domestic Carbon Model (UKDCM) and DECarb, when run with the 40% house scenario, to see the effect of the different demolition scenarios.

First of all, the results from using the classic scenario (demolition scenario 1) were docked with the results from DECarb and the UKDCM, as confidence had already been established in using this demolition scenario. The results produced using the ABM stock transformation method are, as during the back-cast, reasonably similar to those produced by DECarb, as shown in figure 25. The curve produced by UKDCM is an entirely different shape to those produced by the other two techniques, and although the results are similar in places, there are some major discrepancies, particularly at 2010 and 2050, this just comes down to differences in the models.

The population modelled by the ABM is slightly higher than that modelled by DECarb, as demonstrated in figure 27. In order to see how much difference this made the output was manipulated by scaling down the energy consumption by the same amount that DECarb's final population is scaled down from the ABM's final population, the results of which can be seen in figure 26. The adjusted curve is pretty similar to that produced by DECarb and lends further credibility to the results.

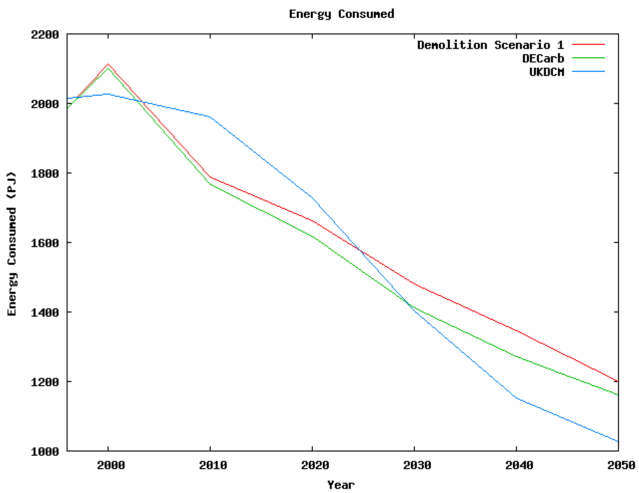

Figure 25: 40% House Energy Consumption

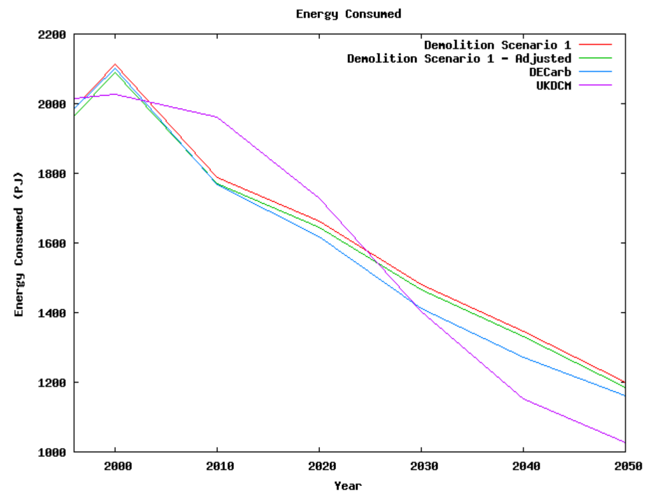

Figure 26: Adjusted Energy Consumption

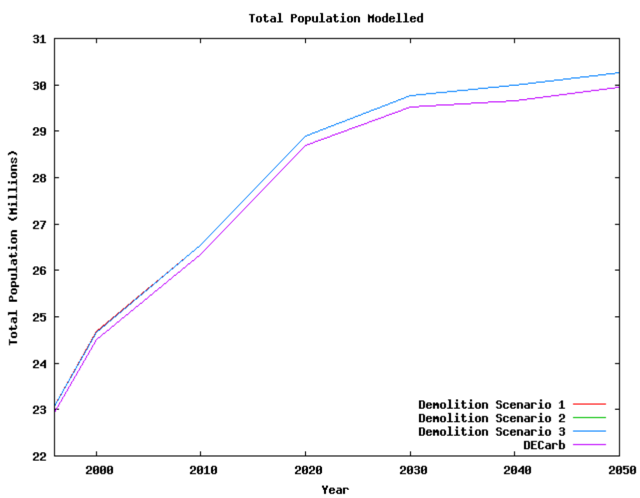

Figure 27: Total Population Modelled

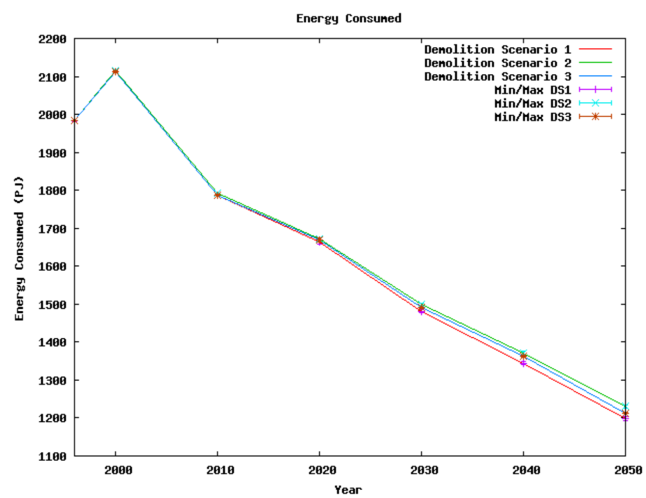

Figure 28: Demolition Scenarios

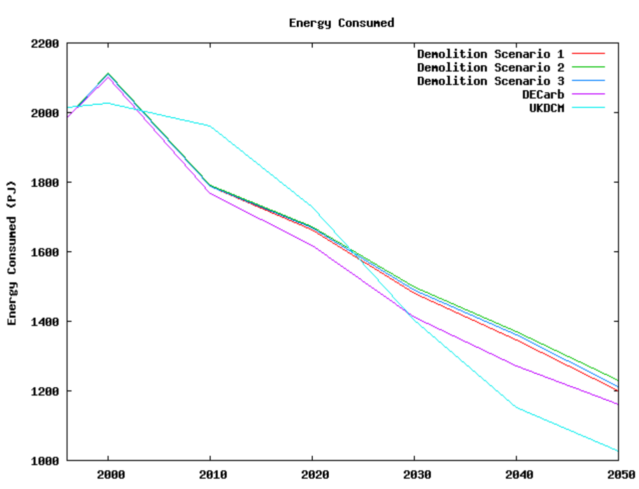

Figure 29: Demolition Scenarios against DECarb

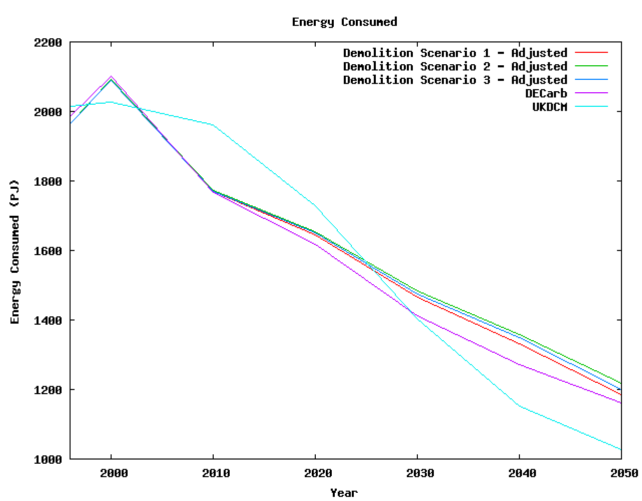

Figure 30: Adjusted Demolition Scenarios

In order to see how much of an impact using the different demolition scenarios would have, all three scenarios were run with the 40% House conditions, as shown in figure 29. Up until 2010 all 3 scenarios produce similar results, but from 2010 onwards their results deviate slightly. Demolishing the oldest dwellings results in the lowest carbon emissions, whilst demolishing those dwellings with the least insulation results in slightly higher energy consumption. In this case, how similar the ABM results are to those of DECarb and UKDCM is a secondary concern. What is of more interest is how the different demolition scenarios actually have a visible impact on the predicted energy consumption. The figures from this graph are summarised in table 10.

In 2050 the difference between the highest and lowest predicted energy consumptions is over 40 petajoules (PJ), which is a considerable difference. Another interesting aspect is the consistent range of results produced by demolition scenario 3. The difference between the minimum and maximum predicted values in this scenario are, with the exception of 2020, consistently smaller than the those seen in the other two scenarios. This exhibits the trade mark of emergent behaviour, as the results appear too consistent to be random.

Table 10: Energy Consumption for 2020 – 2050 (PJ)

Demo scenario	**2020 Ave**	2020 Max	2020 Min	**2030 Ave**	2030 Max	2030 Min	**2040 Ave**	2040 Max	2040 Min	**2050 Ave**	2050 Max	2050 Min
DS1	**1662**	1667	1658	**1480**	1485	1476	**1344**	1349	1340	**1198**	1203	1193
DS2	**1671**	1673	1667	**1498**	1501	1493	**1371**	1375	1367	**1231**	1234	1226
DS3	**1669**	1673	1664	**1490**	1491	1487	**1362**	1364	1360	**1213**	1215	1212

When measuring carbon emissions the results were in a similar vein. No carbon data could be found from UKDCM, so figures were solely measured against those of DECarb. The deviation from emissions predicted by DECarb was slightly greater than the deviation experienced when measuring energy consumption, even taking into account the adjustment for population size (see figure 24).

7.6 Conclusion

This cycle of the project explored what effect using different demolition scenarios would have upon the total carbon emissions and energy consumption of the model. To carry out this task three main scenarios were outlined and a simple metric was devised to calculate how suitable a dwelling would be for demolition.

Using the forecast scenario specified in the 40% House project, the ABM stock transformation model was docked both with DECarb and UKDCM. Results were found to follow a similar curve to those predicted by DECarb and although the deviation from the UKDCM results was considerate, these are only predictions and so can not be verified.

The same experiment was carried out with all 3 demolition scenarios and it was found that using each one caused noticeable differences in the total carbon emissions and energy consumption calculated. The value of this is that it confirms the fact that a demolition scenario can have a reasonable influence on the figures calculated, and this fact advocates the need for a more accurate demolition scenario in future models.

Chapter 8

Cycle 4 - Implementing Behaviours

The final extension carried out to the model was implementing a behavioural framework for the agents. As discussed in section 5.2, in order to dock successfully with Natarajan's EBM, using marionettes was the simplest way forward. Given that initial confidence in the model had been developed, it was felt it would be interesting to see how else the model could be applied. This section explores replacing the marionettes with what Gulyas describes as bounded rational agents (see section 2.4.2). On Gulyas's scale of where different implementation techniques lie between being EBMs and being ABMs, systems using marionettes lie on the EBM side of the middle, while systems using bounded rational agents are considered to be fully agent-based.

Both DECarb and the existing ABM only acted on user defined values, there was no explicit behaviour being modelled. In order to keep this scenario as simple as possible, only one household characteristic was chosen to look at - window insulation. The aim was to implement some sort of behavioural framework from which households could infer whether they should obtain window insulation, as opposed to this decision being made for them at a global level. The first task was to create a behavioural model that could be implemented using the existing ABM.

8.1 Defining the Behavioural Model

As was shown in section 2.4.2 there is a frustrating lack of research looking specifically at energy-related purchasing behaviour. The majority of research that has been done does not build constructively on past work, and little common ground can be derived from the papers. In ideal circumstances, a paper defining the different types of energy-related purchasers and how they react to external stimuli would have been available. Unfortunately, to the author's knowledge, no such framework exists, so it was necessary to create one from the available literature on the topic.

8.2 Using Van Raaij and Verhallen's Behavioural Framework

Although Van Raaij and Verhallen have carried out extensive work on energy-related behaviours, their focus has always erred towards energy-saving behaviours rather than purchase-related behaviours. As a result of this a purchase-related behavioural model needs to be derived. Their *Behavioural Model of Residential Energy Use*, as seen in figure 1, contains many external factors, the combined effect of which controls residential energy use, aspects of this model could be used to model purchase-related energy behaviour. Trying to model all of these factors in one ABM is far beyond the scope of this project, and arguably, not necessary. As discussed in section 2.4.3, a simulation should only aim to model the core dynamics of the system, a complete description of the problem is unnecessary.

What follows is a discussion attempting to identify components of the behavioural model which are, primarily, part of the core dynamics of the system, but, secondly, feasible to model.

8.2.1 Energy-Related Behaviour

This module is core to the model and dictates how a household reacts when faced with making an energy-related decision. Three type of energy-related behaviour are discussed; purchase-, usage-, and maintenance-Related. Whilst usage- and maintenance-related behaviours are not of much interest when trying to model the uptake of energy-related technologies, purchase-related behaviour is.

Purchase-related behaviour equates to exactly the kind of behavioural trait being looked for, and it is implied that the set of the population that falls into this behavioural category are those who would most readily adopt an energy saving technology.

To accurately model these behaviours using agents would require more empirical research in order to find what proportion of the population falls into which category, and further details on how each category behaves. But these categories are a good starting point for an experimental model of energy-related behaviour.

8.2.2 Characteristics of the Home and Appliances

Van Raaij and Verhallen define 5 characteristics that have the most bearing on energy-related behaviour, those are:

- Home Insulation
- Home Attachment
- Energy Use of Neighbours
- Wind Orientation
- Temperature Difference Between First and Second Floors

The first three characteristics are all modelled in the ABM and as such could easily be taken into account when a household was making an energy-related decision. Again, a problem is that the paper does not go into detail as to how these factors interact to influence behaviour, so further empirical analysis would be required. Household insulation would need to be calculated as a combination of the different insulation characteristics, while attachment is recorded in the dwelling type characteristic with different values representing whether a household is a detached house or a terraced house, and so on. The energy use of neighbours can be obtained by observing the characteristics of different dwellings within some defined 'neighbourhood' of agents.

Wind orientation would be difficult to simulate with the current ABM. Assuming that some kind of meteorological data is obtained on wind orientation, it would still be necessary to entirely change how the model is populated. At the moment, agents are placed randomly on a 2D grid, so some additional structure would be required. For example, if the concept of wind direction was modelled, then it seems that agents would have to be distributed not randomly, but in a pattern representing the actual dwelling pattern of the region.

8.2.3 Sociodemographic Factors

Although sociodemographic factors are not currently modelled, some of those considered are feasible to implement in the ABM.

Van Raaij and Verhallen rank household income as one of the most important factors in determining household energy behaviour. They also document various studies providing evidence of households in different income bands uptaking new technologies with different propensities. For example middle-income households are found to be more likely to adopt a new technology than low-income households, who cannot afford to adopt, and high-income households, who can afford to not adopt. Ignoring the fact that there will be some correlation between a household's characteristics and the income band of a household, this is a situation that could be modelled by assigning at random an income band to every household created. The final factor to consider is whether the initial investment can be recouped instantly, or if it is a long term investment. Those in a more comfortable financial position will be able to make long term investments, knowing they will eventually recoup their initial outlay. Those in the lowest income band will rarely invest in a technology unless savings are very short term, as any long term outlay may not be affordable.

Regional differences are considered to be another influential sociodemographic factor, as houses in the north (both northern countries and northern regions of countries) have a higher energy use than those in the south due to colder temperatures[1]. Region is an attribute recorded in the ABM and could be used to influence how likely a household is to adopt a technology.

8.2.4 Energy Prices and Policies

Energy prices and policies are also found to have an effect on energy-related behaviour, as with sociodemographic factors, the effect of these is not currently studied in the ABM but there is definitely scope for its introduction.

The flexibility of an ABM would allow the global broadcast of a change in energy prices, or a new government policy at any point in the simulation. This additional information could alter the propensity by which an agent is likely to adopt a new technology. For instance, lower energy prices could mean a household is less likely to purchase double-glazing, whilst a government policy offering double-glazing subsidies could increase the likelihood that a household will adopt double-glazing.

8.2.5 Climate, Season and Weather

A final aspect of van Raaij and Verhallen's model that could be used is the effect of climate, season and weather. Currently the user chooses a climate profile when defining their scenario, but this profile is only used for energy calculations. The weather is, currently, completely unrelated to modelling the housing stock, so this could be introduced with no changes to the existing model required. As with policy and energy prices, the weather could be broadcast globally to all agents at every step, this could then be taken into account in their behaviour.

8.2.6 Broadcasting Global Events

If events such as policy changes and weather patterns were broadcast globally, the agents could still be regarded as bounded rational agents, because the information that they were receiving on a global level would still be being modelled consistently with the real-world situation. This global information received would then be taken into account with their local 'bounded' information to produce their perception of the world.

1 Presumably just in the northern hemisphere

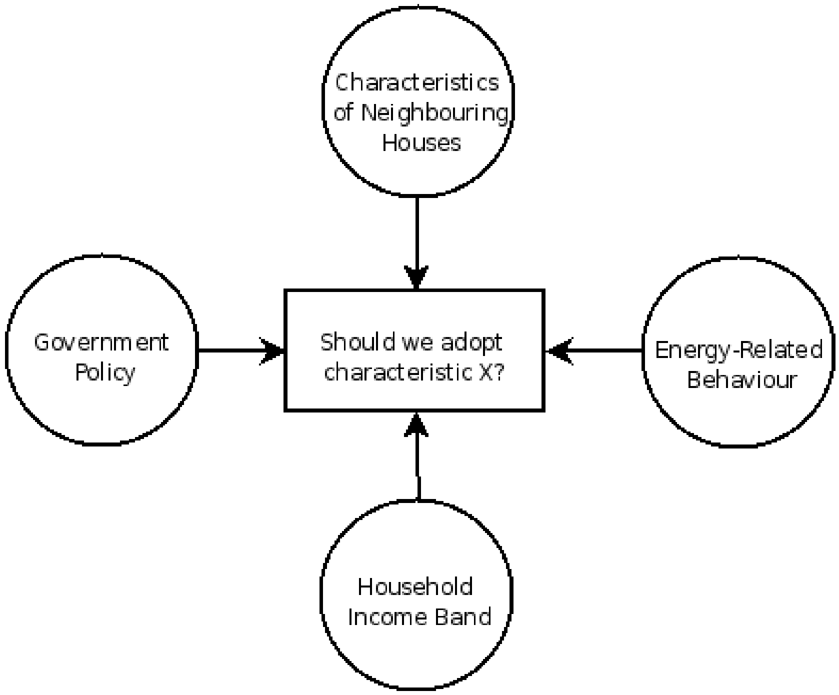

Figure 31: Simple behavioural framework regarding energy-related purchasing behaviour

8.3 Designing a Simple Behavioural Framework

After considering the core components of van Raaij and Verhallen's model, a simple behavioural framework was created, as shown in figure 31. For the purposes of assessing the suitability of this technique, only window insulation was looked at.

The household's standard decision making process is based on how many of its neighbours have window insulation. So if a certain proportion of households in a household's neighbourhood have window insulation, they will get it as well. This should produce a gradual rise in uptake which will allow the assessment of how influential other factors in the model are.

Every household is assigned one of the three aforementioned energy-related behaviours. The propensity with which those households with purchase-related behaviour will uptake a technology is greater than that of those with maintenance- and usage-related behaviour.

The household's income band will also affect whether they uptake adopt a technology. It has been shown that those households in the middle income band will adopt first and those in other income bands will adopt later, if it all.

The final factor in the model is government policy. This will allow exploration in to how uptake is effected by a 'stochastic shock', an external event which may alter the outcome of a household's decision making process.

8.3.1 Calculating Income Band Data

Verhage (1980) carried out a study showing that early adopters of new energy saving technologies are most highly represented by mid income earners, the results are summarised in table 11.

Table 11: Findings from Verhage (1980) regarding uptake of energy-related technology

	Low Income (< 25k guilder)	Mid Income (> 25k and < 50k guilder	High Income (> 50k guilder)
Early Adopters	20%	57%	23%
Late Adopters	12%	37%	51%

In order to effectively make use of this data, it was first necessary to convert these earning in Dutch guilder (denoted f) from 1980 to their equivalent current day values in British pounds (GBP) .

In order to approximate the equivalent worth in GBP of f1in 1980 the historical exchange rates were used. In 1980 1 American dollar (denoted $) was worth f1.9875 and £0.4299. Therefore making f1 worth $0.503 and £0.22 (Measuring Worth, 2007a). £0.22 in wages in 1980 was found to be the equivalent of £1.02 in 2008, leaving f25,000 in 1980 being worth £25,500 in 2007 (Measuring Worth, 2007b).

Data provided by the office for national statistics (National Statistics Online, 2007) shows that in the UK in 2005/6 about 3/5 of households had an income of under £25,500, while 1/5 had an income between £25,500 and £51,000 and 1/5 had an income over £51,000.

8.4 Implementing the Simplified Framework

8.4.1 Changes in AgentFactory

Changes were made in AgentFactory in order to assign 3/5 of agents with an attribute marking them as low-income band, and 1/5 each with mid- and high-income bands. Each of the 3 behavioural characteristics (purchase-, maintenance- and usage-related) were distributed evenly – one per agent – amongst the population.

8.4.2 Parameter Sweeps

To allow experimentation with the behavioural model, several additional parameters were added to the ABM, these are detailed in table 12.

Defining these additional parameters provided the functionality of performing various parameter sweeps. Parameter sweeps are carried out in order to see the effect of altering a value whilst keeping all others the same.

Table 12: Parameters added to the ABM for behavioural experiments

Parameter	Type	Task
Length Of Agent Vision	int n	Sets neighbourhood size to n x n
Influence Of Neighbours	int n	The lower the number the more influenced a household is by its neighbour
Year To Start Scheme	int n	The year to start a government scheme
Length Of Gov Scheme	int n	The duration of the government scheme
Early adoption threshold	int n	Adoption in the first n years is considered early

It is important that parameter sweeps are carried out in experimental conditions, with particular attention being paid to the need to keep all other parameters at a constant level when testing values over one particular parameter.

8.5 Experiments

8.5.1 Experimental Conditions

The aim of carrying out each experiment was to gain knowledge about the variable being changed. In order to make this effective it was important to make sure that all other variables were set at a constant value. Every experiment was carried out ten times, and figures shown are always the average of all 10 runs. To gain more confidence in the results, carrying out more than 10 runs would be necessary, but the main aim of this section is to show how the behavioural model *could* be used, rather than conclusively proving any statements.

Each experiment had an associated experimental hypothesis. This is a claim that the successful execution will either be able to prove or dispel. First defining a hypothesis gives a clear purpose to an experiment and is useful as a reference point when selecting parameters to use.

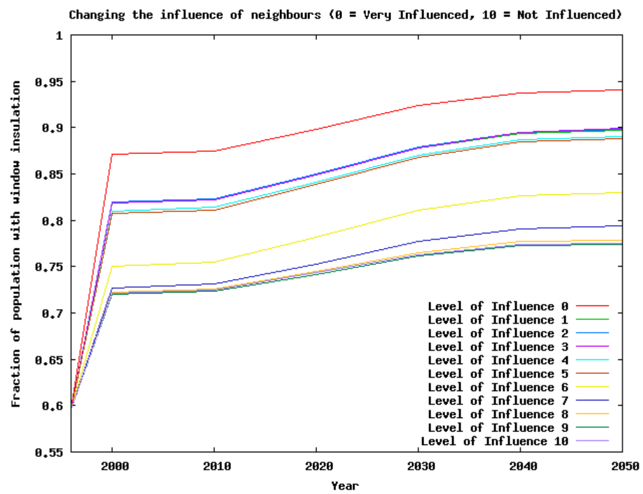

Figure 32: Parameter Sweep on how influential neighbours are

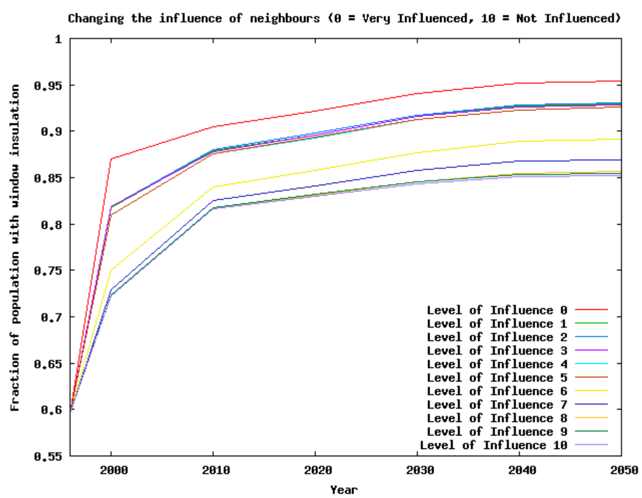

Figure 33: Parameter sweep on how influential neighbours are with a 3 year government scheme in 2000

8.5.2 Experiment 1 – Exploring the Influence of Neighbours

The experimental hypothesis was "*A population of households more influenced by their neighbour's decisions will have a higher uptake of window insulation*". The experiment carried out was a simple parameter sweep on the range of values available defining how influential a household's neighbours are. Setting the parameter equal to 0 results in all households being very influenced by their peers, with even the sight of a small fraction of neighbours with window insulation enough for a household to choose to adopt. A value of 10 results in all households having to see the majority of their neighbours with window insulation before they themselves adopt it.

The results, as seen in figure 32, are as expected. The more influenced each household is by its neighbour, the higher the global uptake of window insulation. In all cases the initial uptake was very sharp and then smoothed out between 2000 and 2010. This quick piece of face validation shows that the framework behaves as expected in a simple scenario, and adds credibility to results in future parameter sweeps. This experiment provides us with a base case which can be used to test the effects of other changes to the environment.

This experiment was repeated, but with a government policy introduced in 2000. The scenario was run with the government policy lasting 1,2 and 3 years in order to see what the overall effect on the population was. All three experiments provided similar shaped graphs, figure 33 shows the output for the 3 year scheme (graphs for 1 and 2 year schemes can be found in appendix D, figures 45 and 46 respectively). The main difference between figures 32 and 33 is the period between 2000 and 2010, with the government scheme resulting in quite a sharp uptake in this period, as opposed to a slow uptake in the same period without the scheme. Introducing the government scheme also results in even the households least influenced by their neighbours seeing a 0.05 greater uptake then their corresponding households that were not exposed to the scheme. The effects of 1 and 2 year schemes appear to be very similar, with a 3 scheme appearing to be marginally more effective. This is an interesting aspect to explore in the next experiment.

8.5.3 Experiment 2 – The Effect of Governmental Schemes

Being able to accurately predict the effect of a policy before it is released would be an ideal scenario for a policy maker. One potential use of ABM is to model the way in which the housing stock may react to the introduction of new energy-related policies. A simple experiment is carried out here where a hypothetical government scheme is introduced, promoting the uptake of window insulation.

Working with an experimental hypothesis of "*The longer the length of a government scheme promoting window insulation lasts, the higher the total adoption rate will be*", the experiment involved carrying out a parameter sweep on the duration of the government scheme. The variable determining the influence of neighbours upon a household was set to a mid-range value of 6, this was to ensure that adoption rate was neither too conservative or too frivolous.

The results, as seen in figure 34, provide some interesting talking points. As predicted, the effect of introducing a government scheme was that it increased the global uptake of window insulation. But when looking at the effect of the length of the scheme, the results were not so clear cut.

A 1 year scheme clearly increases the uptake from a scenario where no scheme was

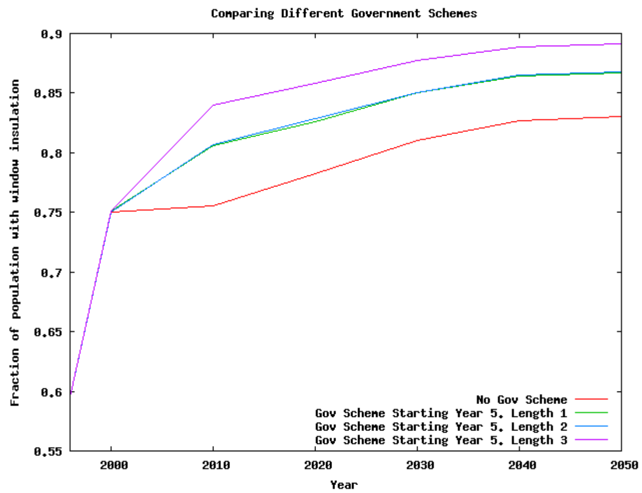

Figure 34: Comparing the effect of different lengths of government schemes

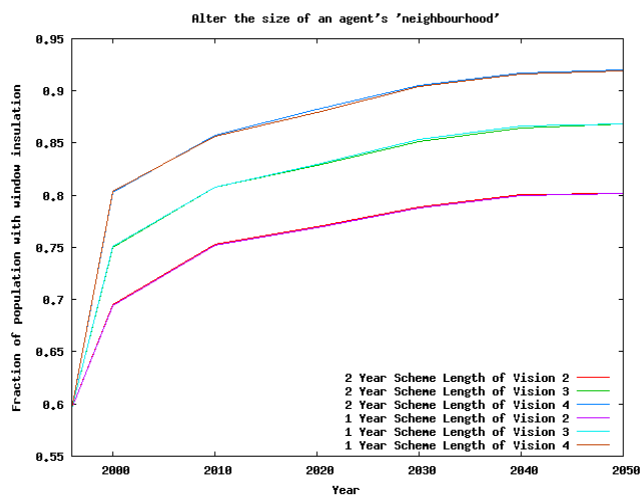

Figure 35: Parameter sweep on the size of an agent's neighbourhood

introduced, but a 2 year scheme appears to have no additional effect on uptake. A third run of the model with a 3 year scheme provided a noticeably higher global uptake than the 1 and 2 year schemes. This behaviour can be explained by the fact the early adoption threshold (see table 12) was set to 2. This explained the additional uptake in year 3, as later in the life time of a scheme, high- and low-income band households also adopt the technology. Those households in a high-income band will adopt because they have seen the success of the scheme, but could afford to take their time. Low-income households may adopt if they can recoup their investment over a short period, but were not confident enough to make the initial investment.

8.5.4 Experiment 3 – Altering the Size of an Agent's Neighbourhood

In experiment 2, the effect of the 1 and 2 year schemes was very similar. In the third experiment the effect of changing the size of what a household regards as its neighbourhood was explored, in order to see if it increased the effectiveness of the 2 year scheme over the 1 year scheme.

In previous experiments, a neighbourhood was regarded as a 3 x 3 area of the 2D grid. Here the effect of considering a neighbourhood as a 2 x 2 grid and a 4 x 4 grid was compared against the behaviour observed with 3 x 3 neighbourhoods.

The results were as expected, in that, the larger the area an agent considered its neighbourhood, the higher the global uptake of window insulation, this is shown in figure 35. The reason for this is that if every agent has a larger neighbourhood, there is a higher probability that ones of its 'neighbours' will adopt window insulation. The results again showed no visible difference between the uptake of window insulation observed in 1 and 2 year schemes. The seems to back up the theory that this the similar results of these to variables was down to the early adoption threshold being set as 2. In order to test this, further experiments carrying out a parameter sweep on the early adoption threshold would be necessary.

8.6 Conclusion

The aim of creating and implementing a simple behavioural model was not to try and discover new global trends in energy-related behaviour. Although, eventually, this would be the goal of this style of modelling, there is still a long way to go. This section demonstrated the kind of experiments that could be carried out with a population of heterogeneous agents. The use of parameter sweeps was demonstrated as technique for tweaking the values used to control agent behaviour, and a simple experiment exploring the introduction of government legislation was presented as a potential area of use for ABM. Before any conclusive work of this type is carried out it would first be necessary to obtain a more complete behavioural framework to base an ABM on.

Chapter 9

Conclusions

9.1 Findings

This project has demonstrated the feasibility of using ABM to model the transformation of the UK housing stock. Building on the work done by Natarajan & Levermore (2007a) on DECarb, an ABM stock transformation method has been integrated with their existing front- and back-end. Initially the ABM was created with a population of marionette agents, which had their behaviour defined at a global level.

When run in a back-cast scenario the ABM stock transformation method produced results within -2.65% of the actual historical data when predicting carbon emissions and 2.51% when predicting energy consumption. These results are close enough to confirm the feasibility of using ABM in this scenario.

The value of using an ABM stock transformation method was further highlighted when considering extensions to the model. Scenarios that would have been very hard to explore using the existing EBM stock transformation method became much more accessible due to the flexibility offered by modelling with agents. The first facet of the model explored was that of demolition. Two new ways in which to handle demolition were created to complement the existing method of demolishing the oldest dwellings. One technique looked at demolishing random dwellings from the stock, and the other demolished dwellings chosen by using a simple metric to determine those worst insulated. Results from this experiment demonstrated the impact that different demolition scenarios can have on the output of the model, and highlighted this as an area that could be modelled more accurately in the future.

The final area of the domain explored was that of energy-related behaviours. Firstly components of van Raaij and Verhallen's (1981) model of energy related behaviour were discussed both in terms of how how core they are to the model and also how feasible each is to model using agents. Four key components; energy-related behaviour of the household, the income-band of the household, energy-related behaviour of neighbours and government policy, were chosen as constituent parts of a simple behavioural framework to use with the existing ABM. Implementing this framework created a population of heterogeneous agents, another advantage offered over EBM which treats the population as a set homogeneous entities. A few simple experiments were carried out using this behavioural framework in order to show the possibilities offered by modelling in this way. First it was shown how parameter sweeps can offer a way to explore difference scenarios, and a base case for window insulation adoption was created based on the actions of a household's neighbours. An experiment was then carried out to show the difference government policies can make in adoption levels, and further experiments were carried out in order to observe the effect of changing the length of government policies.

The suitability of ABM for the transformation of the UK housing stock has been demonstrated, and an idea of how energy-related behaviours could be used has been explored. An important aspect of using ABM has also been shown to be its flexibility, as seen by experimenting with demolition scenarios, a situation that would be very complex to explore using an EBM. The potential of using ABM in the energy domain has been demonstrated, and there are many avenues that could be explored using the existing model.

9.2 Future Work

It has been shown that using marionettes to model the UK housing stock will give feasible results, but to achieve any additional degree of accuracy, household behaviours need to be taken into account. Treating the population as a group of homogeneous marionettes has been explored as far as possible. It is necessary to delve deeper into the behaviour of the constituent members of population in order to accurately model their energy-related actions, and the resulting emergent behaviour of the nation as a whole.

Exploring the Effect of Weather and Climate Change

Currently in DECarb the climate is taken into account when calculating carbon emissions and energy consumption, but it is not used to influence the stock transformation method. Initially, simple assumptions could be made as to how a household may behave if the climate changed, for example a household may be less likely to purchase any kind of new energy-related technologies if the climate becomes warmer.

Zero-carbon Dwellings by 2016

Under legislation recently released by the UK government, all new homes will need to be zero-carbon by 2016 (Building a greener future, 2006). There are several difference scenarios that the model could be used to explore. Firstly the effect on total carbon emissions and energy consumption could be modelled by introducing zero-carbon new dwellings at different rates, or possibly a user-defined rate. Secondly, the knock-on effect of this legislation could be explored. For instance, the behaviour of house owners in response to new dwellings becoming zero-carbon may be to make their own dwellings zero-carbon in order to maintain the value of their property.

In both of these scenario it would first be necessary to create a metric judging whether a house is zero-carbon, based only on the core attributes that are currently modelled. Once this metric was devised, experimenting with aforementioned scenarios would not be overly complex.

A More Accurate Demolition Model

Another area with a huge amount of potential is that of demolition scenarios. It has been shown in this project that the demolition scenario used can make a noticeable difference at a macro level. A more theory-based model to predict demolition would grant an additional level of credibility to any ABM of the housing stock.

Ideally this model could be implemented using the existing data streams into the ABM, but as these were not chosen with this scenario in mind it may be necessary to obtain additional data.

Dissemination of Information

An aspect of ABM not at all touched on in this project is the social ability of agents. Although in later experiments, agents are aware of their neighbours, and indeed take heed of their actions, there is no notion of communication. Van Raaij and Verhallen (1981) acknowledge that visible forms of energy conservation do have a greater effect, but still stress the importance that dissemination of energy-related information through a social network can have. Their paper comments on the lack of research done on the role of social interactions and the community in stimulating the adoption of energy-conserving home improvements, although this criticism was made over thirty years ago, no literature has been discovered suggesting that the situation has changed dramatically.

Wooldridge (2002) lists social ability as one of the three main characteristics of an intelligent agent, and as such an ABM would be perfectly suited to a scenario where the dissemination of information needed to be modelled.

Changing Behaviours

Another aspect of Van Raaij and Verhallen's (1981) model is energy related attitudes. They found that although some people have energy-conscious attitudes, they do not behave in an energy-conserving way. They put this down to a lack of knowledge.

An extension of looking at dissemination of information is seeing how energy-related knowledge travels around social networks, and how global broadcasts containing energy-related knowledge can further disseminate beyond the initial recipients. The effect of this can then be ascertained by observing how energy-related behaviours of households change over time.

Learning from Past Experiences

An important factor not yet considered is that of learning. Learning is a task intrinsically carried out by humans, and previous experience is considered a vitally important component of human decision making (e.g. Anderson, 1990, 1993). Vriend (2000) classifies two different modes of modelling learning; social and individual. To accurately model the behaviours of the households in the UK housing stock, it would be important to treat learning as an individual task, that is, an agent learns exclusively on the basis of its own experiences.

If applicable research could be found it could then be applied for simple scenarios such as households in a low income band adopting a technology with less chance if they had made a purchase recently. More complicated scenarios that could be explored involve the agents learning from the effects of previous purchases, i.e. if a previous purchase did not offer as much of a saving as hoped, then a household may be more apprehensive about future purchases, and learning from the past experiences of neighbours, which would again require the agents to communicate.

Looking at How Different Behavioural Groups React

Van Raaij and Verhallen (1982) categorise households into 5 different categories of energy-consumption behaviour; Conservers, Spenders, Cool, Warm and Average. The categories are defined only by their energy-consumption behaviour, and as such are of no direct use in the existing model, as energy consumption is not modelled at all.

An interesting experiment would be to see how different categories of household energy consumption react to government legislation. Modelling this would require some information beyond household energy consumption on each of these groups. One viable technique would be to cross-reference a household's consumption behaviour, with some other scale defining purchasing behaviour.

Modelling the Effect of Upgrading Efficiency Before Sale

In the new Home Information Packs (HIP), an energy-efficiency rating system, ranging from G to A, has been created. Boardman (Natarajan, 2008) has proposed legislation regulating the sale of properties. Before being sold, any dwelling below a certain level of efficiency, would have to show improvement enough to be classed in the next level of efficiency. The model could be used to see how long it would take before all of the stock reached a certain efficiency level.

A metric would have to be devised in order to calculate which efficiency class a household falls in to, but, primarily at least, this could be a simplified metric based on the actual categorisation. Some data on sales figures would also be required in order to model the sales rates more accurately.

This is a good example of how the model could be used to explore legislation, and would be reasonably simple to implement. Assuming sales data was available, the only non-trivial element would be devising the metric to categorise households.

9.3 Critique of Work Carried Out

When the literature survey was carried out, the project had two primary aims that could have been carried out in a number of ways. A lot of the literature regarding ABM focuses on the use of what Gulyas (2005) classes as bounded rational agents, and this seems to be regarded as the traditional use of agents. As such, a lot of the literature survey focuses on this flavour of ABM. Although these findings were not directly related to the majority of work done in the first 3 project cycles, they shaped the long term aim of implementing behaviours in the model. This research also emphasised the fact that the primary aims of this project could not be met using traditional ABM techniques, and that a less traditional approach, taking aspects from both EBM and ABM, was the ideal solution. Gulyas (2005) and Parunak et al. (1998) both briefly mention that in some situations, this hybrid approach is the ideal choice in order to model the problem satisfactorily.

The implementation was carried out successfully and coding has been done in a scalable manner in order to allow for future extensions to the model. The only current uncompleted part of the implementation is that of accounting for different regions. Currently, one run of the model only accounts for one region. Because the four regions are all effectively independent of each other, each one can be run in a different space. Rather than introducing the complexity of four spaces in the model, the region to observe was left as a user-defined parameter, it was felt that time could be better spent exploring other scenarios than integrating these spaces. The current situation does not detract from or effect the results of the simulation, but just results in a slightly extended running time.

When tested in the back-cast scenario, the ABM using marionettes provided very similar looking results to the EBM of DECarb, and this allowed for a successful docking of the two systems. Although these results were obtained using agents and were very convincing in their accuracy, the marionette implementation was not strikingly different to the existing EBM. The method used is still very statistical and just a slightly different way of achieving the same goal. On the other hand, one important aspect of this technique is that it does attempt to model every single household defined in the data for 1996, and therefore every simulation offers a hypothetical scenario as to exactly how the stock will behave. Although their methods may be similar, the marionette-based ABM stock transformation method appears to be more credible than the EBM method. This is partly down to the fact that every household is represented as an object rather than a number in a matrix, meaning that the results cannot be manipulated in the same way as in the EBM. The technique used in Natarajan & Levermore's (2007a) stock transformation method where any negative values are equally distributed over the population in order to remove them is a perfect example of this.

The work carried out on demolition scenarios was done well and provided interesting results, this is certainly an aspect of the model that would be much harder to explore using EBM. The only possible extension to this scenario would have been to explore demolition scenarios using metrics other than the one devised. The usefulness of this would have been limited, as the metric created, and any future metrics, had bases only in common-sense and guesswork, and without a more concrete model to implement, little more could be explored here.

The aspect of this project which leaves the largest scope for future work is the work done on energy-related behaviours. The reason that this facet was explored last was primarily due to the lack of available research in the area. In fact the sparseness of literature available was most ably highlighted by the fact that the simple behavioural model created is primarily based on research from almost 30 years ago (Van Raaij and Verhallen, 1981). Although Keirstead (2005) champions the idea of using Van Raaij and Verhallen's model for ABM, his paper is rooted firmly in the energy domain and no implementation details are proposed. For this reason, the first part of the section discusses how different modules of Van Raaij and Verhallen's (1981) model could be implemented in the existing ABM, due to time constraints only parts of the model that would be feasible to implement without a lot of additional work were discussed. Section 9.2 discusses some larger extensions based on Van Raaij and Verhallen's model, that would not be so trivial to implement in the existing ABM. After identifying key aspects, a simple behavioural model was constructed based on components which are core, but also fit inside the existing framework of the ABM. From a review of the available literature this simple model appears to be novel within the domain, and ideally demonstrates the potential uses of behavioural modelling. The work done on energy-related behaviours was severely constrained by the lack of available behavioural models, and more served to show the possible uses in the domain of ABM using bounded rational agents.

9.4 Summary

This project has demonstrated how ABM can be used as a stock transformation method in order to forecast carbon emissions and energy consumption. This method has been validated using back-casting against actual data from 1996-1970, results were within -2.65% of the historical figures for carbon emissions, and within 2.51% of the actual figures for energy consumption. An exploration of demolition scenarios has demonstrated that using different scenarios has a bearing on the outcome of the simulation, and identified this as an area where further work needs to be done. Finally a review of Van Raaij and Verhallen's *Behavioural Model of Residential Energy Use* resulted in the creation of a novel simple behavioural model tailored to the study of purchase-related behaviours using ABM. An important point on which to end this project is one raised by an ABM expert at a UK Energy Research Centre discussion on the applications of ABM.

"there is often an expectation that ABM is a substitute for technology, economic or market models. ABM is not a substitute, it is a simply a different kind of model" (Keay-Bright, 2007)

An ABM requires some theoretical grounding, it cannot be created out of nothing. This area of huge potential cannot be fully utilised until further empirical research is done to in order better understand and categorise purchase-related behaviours that households exhibit.

Chapter 10

Bibliography

Agile Alliance, 2001. *Manifesto for Agile Software Development*. Available from http://agilemanifesto.org/ [Accessed March 21 2008].

Anderson, J., 1990. *The Adaptive Character of Thought*. Lawrence Erlbaum.

Anderson, J., 1993. *Rules of the Mind*. Lawrence Erlbaum.

Balci, O., 1998. Book Chapter: Verification, Validation and Testing. *In Handbook of Simulation: Principles, Methodology, Advances, Applications and Practice*. New York: John Wiley & Sons.

Barr, S., Gilg, A.W. and Ford, N., 2005. *The household energy gap: examining the divide between habitual- and purchase-related conservation behaviours.* Exeter: Department of Geography, University of Exeter.

Bender, B., 1979. The Determinants of Housing Demolition and Abandonment. *Southern Economic Journal*, 46(1), pp. 131-144.

Berry, B.J.L., Kiel, L.D., and Elliott, E., 2002. Adaptive agents, intelligence, and organization: Capturing complexity in agent-based modeling. *PNAS,* 99(3), pp. 7187-7188.

Black, J.S., Stern, P. and Elworth, J.T., 1985. "Personal and contextual influences on household energy adaptations. *Journal of Applied Psychology 70(1)*, pp. 3-21.

Bonabeau, E., 2002. *Agent-based modeling: Methods and techniques for simulating human systems.* Cambridge: Icosystem Corporation, 545 Concord Avenue.

Bordini, R.H., Fisher, M., Visser, W. and Wooldridge, M., 2004. *State-Space Reduction*

Techniques in Agent Verification. Liverpool: University of Liverpool.

Brueckner, S.A. and Parunak, H.V.D., 2003. Resource-Aware Exploration of the Emergent Dynamics of Simulated Systems. *In: AAMAS '03: Proceedings of the second international joint conference on Autonomous agents and multiagent systems.* New York: ACM, pp. 781-788.

Cockburn, A., 2002. *Agile software development*. Boston: Addison-Wesley.

Collections (Java 2 Platform SE v1.4.2) [online]. Sun Microsystems. Available from: http://java.sun.com/j2se/1.4.2/docs/api/java/util/Collections.html#sort(java.util.List) [Accessed April 18, 2008].

Collier, N., 2002. *RePast: An Extensible Framework for Agent Simulation.* Chicago: Social Science Research Computing, University of Chicago.

Communities and Local Government, 2007. *Building a Greener Future: Towards Zero Carbon Development – Consultation* [online]. Available from: http://www.communities.gov.uk/archived/publications/planningandbuilding/buildinggreener [Accessed April 18 2008].

Communities and Local Government, 2008. *The Code for Sustainable Homes: Setting the standard in sustainability for new homes* [online]. Available from: http://www.communities.gov.uk/publications/planningandbuilding/codesustainabilitystandards [Accessed April 20 2008].

Department For Trade and Industry, 2003. *Energy White Paper* [online]. *Available from:* http://www.dti.gov.uk/energy/policy-strategy/energy-white-paper-2003/page21223.html [Accessed 5 November 2007]

Dillman, D.A., Ross, E.A. and Dillman, J.J., 1983. Lifestyle and home energy conservation in the United States: the poor accept lifestyle cutbacks while the wealthy invest in conservation. *Journal of Economic Psychology 3*, pp. 299-315.

Epstein, J.M. and Axtell, R., 1996. *Growing Artifical Societies.* The Brookings Institution Press.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J.M., 1994. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional.

Gilbert, N., 2004. *Agent-based social simulation: dealing with complexity.* Surrey: Centre for Research on Social Simulation, University of Surrey.

Gulyas, L., 2005. *Understanding Emergent Social Phenomena.* Budapest: Computer and Automation Research Institute, Hungarian Academy of Sciences.

Inchiosa, M.E. and Parker, M.T., 2002. Overcoming design and development challenges in agent-based modeling using Ascape. *PNAS,* 99(3), pp. 7304-7308.

JBoss Rules [online], 2007. Available from: http://labs.jboss.com/drools/ [Accessed 8 December 2007].

Jennings, N.R., 2001. An agent-based approach for building complex software systems. *Communications of the ACM*, 44(4), pp.35-41.

Johnston, D., 2003. A Physically-Based Energy And Carbon Dioxide Emission Model of the UK Housing Stock. Leeds: Leeds Metropolitan University.

Johnston, D., Lowe, R., Bell, M., 2005. An exploration of the technical feasibility of achieving CO2 emission reductions in excess of 60% within the UK housing stock by the year 2050. Energy Policy 33, 1643–1659.

Keay-Bright, S., 2007. *Agent Based Modelling: Application to Policy.* Meeting Report, Oxford: UK Energy Research Centre (UKERC).

Keirstead, J., 2005. *Evaluating the applicability of integrated domestic energy consumption frameworks in the U.K.* Oxford: Environmental Change Institute, University of Oxford.

Kennedy, R., Xiang, X., Madey, G. and Cosimano, T.F., 2005. *VERIFICATION AND VALIDATION OF SCIENTIFIC AND ECONOMIC MODELS.* Notre Dame: Department of Computer Science and Engineering, University of Notre Dame, 2005.

Lane, K., Sinden, G., Killip, G., Peacock, G., and Natarajan, S., 2005. *40% HOUSE PROJECT BACKGROUND MATERIAL B: FORESIGHT SCENARIOS.* Tyndall Centre. Available from: www.eci.ox.ac.uk/research/energy/downloads/40house/background_doc_b.pdf [Accessed 20 April 2008]

Marsaglia, G., 1968. *Random Numbers Fall Mainly in the Planes.* Washington: Boeing Research Laboratories.

Measuring Worth, 2007a. *Exchange Rate Between US Dollar and 40 Countries, 1913 – 2005?* [online] Available from: http://www.measuringworth.com/datasets/exchangeglobal/ [Accessed April 19 2008].

Measuring Worth, 2007b. *Relative Value of UK Pounds* [online]. Available from: http://www.measuringworth.com/ukcompare/ [Accessed April 19 2008].

Mitchell, J.C., 2003. *Concepts in Programming Languages*. Cambridge: Cambridge University Press.

Murphy, J.T., 2004. *Building a RePast Model* [online]. Arizona: University of Arizona.

Available from http://www.u.arizona.edu/~jtmurphy/H2R/HowTo00.htm [Accessed 9 April 2008]

Natarajan, S. and Levermore, G., 2007a. *Predicting future UK housing stock and carbon emissions.* Manchester: Built Environment Research Group, School of Mechanical, Aerospace and Civil Engineering, University of Manchester.

Natarajan, S., and Levermore, G., 2007b. *Domestic Futures - Which way to a low-carbon housing stock?* Manchester: Built Environment Research Group, School of Mechanical, Aerospace and Civil Engineering, University of Manchester.

Natarajan, S., (S.Natarajan@bath.ac.uk), 25 April 2008. Email to L.Elliott (lje21@bath.ac.uk)

National Statistics Online, *Household Income* [online], 2007. Available from: http://www.statistics.gov.uk/cci/nugget.asp?id=334 [Accessed April 19 2008].

Odell, J., Parunak, H.V.D., Fleischer, M. and Brueckner, S., 2001. Modelling Agent and their Environment. *In:* Giunchiglia, F., Odell, J., Weiss, G, 2002. *Agent-Oriented Software Engineering III,* Springer, pp. 16-32

Parker, M.T., 2001. What is Ascape and Why Should You Care?. *Journal of Artificial Societies and Social Simulation, 4(1).*

Parunak, H.V.D., Savit, R. and Riolo, R.L., 1998. Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and Users' Guide. *In: Proceedings of Multi-agent systems and Agent-based Simulation (MABS'98).* Springer, pp.10-25.

Rabone, J., 2005. Computing Random Numbers. Complexity, *Science and Society Conference 2005*. Four Delta Consulting Limited.

Random (Java 2 Platform SE v1.4.2) [online]. Sun Microsystems. Available fromp: http://java.sun.com/j2se/1.4.2/docs/api/java/util/Random.html [Accessed April 19 2008].

Ritchie, J.R.B., McGougall, G.H.G. and Claxton, J.D., 1981. Complexities of household energy consumption and conservation. *Journal of Consumer Research 8*, pp. 233-242.

Robinson, P., 2007. *Multi-agent simulation of the dynamics of social exclusion in school choice.* Bath: University of Bath.

Sardianou, E., 2007. *Estimating energy conservation patterns of Greek households.* Athens: Home Economics and Ecology, Harokopio University.

Shah, A.P. and Pritchett, A.R., 2004. Work Environment Analysis: Environment Centric Multi-agent Simulation for Design of Socio-technical Systems. *Joint Workshop MABS 2004.* New York: Springer, pp. 65-77.

Shillingford, N., Madey. G. and Kennednoty, R.C., 2005. *Verification and Validation of an Agent-based Simulation Model.* Notre Dame: Department of Computer Science and Engineering, University of Notre Dame.

Shorrock, L.D. and Dunster, J.E., 1997. *The physically-based model BREHOMES and its use in deriving scenarios for the energy use and carbon dioxide emissions of the UK housing stock.* Watford: Building Research Establishment (BRE), Garston.

Sichman, J.S., 2007. *Multi-Agent-Based Simulation (Mabs)* [online]. Available from: http://www.pcs.usp.br/~mabs/ [Accessed 8 December 2007].

Stern, P., 1992. What psychology knows about energy conservation. *American Psychologist,* 47(10), pp. 1224-1232.

Terna, P., 1998. Simulation Tools for Social Scientists: Building Agent Based Models with SWARM. *Journal of Artificial Societies and Social Simulation,* 1(2).

Tesfatsion, L. and Axelrod, R., 2007. *On-Line Guide for Newcomers to Agent-Based Modeling in the Social Sciences* [online]. Iowa: Department of Economics, Iowa State University. Available from http://www.econ.iastate.edu/tesfatsi/abmread.htm#ComplexABM [Accessed 9 December 2007].

Tesfatsion, L., 2008. *Self-Study Guide for RePastJ – Java Based RePast* [online]. Iowa: Department of Economics, Iowa State University. Available from http://www.econ.iastate.edu/tesfatsi/repastsg.htm. [Accessed 9 April 2008].

The Warm Front Team, 2007. *The Warm Front Scheme* [online]. Available from: http://www.grantsforheating.co.uk/ [Accessed March 22 2008].

University of Buenos Aires, 2007. *Complex Systems at the University of Buenos Aires* [online]. Available from: http://www.cea.uba.ar/aschu/complex.html [Accessed 1 December 2007].

Van Raaij, W. F. and Verhallen, T.M.M., 1982. *A Behavioral Model of Residential Energy Use.* Rotterdam: Erasmus University, Rotterdam.

Van Raaij, W. F, and Verhallen, T.M.M., 1983. *Patterns of Residential Energy Behaviour.* Rotterdam: Erasmus University, Rotterdam.

Verhage, B. J., 1980. Stimulating Energy Conservation: Applying the Business Heritage of Marketing. *European Journal of Marketing*, 14(4).

Vriend, N. J., 2000. An illustration of the essential difference between individual and social learning, and its consequences for computational analyses. *Journal of Economic Dynamics and Control*, 24(1), pp. 1-19.

Wooldridge, M., 2002. *An Introduction to MultiAgent Systems*. John Wiley & Sons, Ltd.

Wooldridge, M. and Jennings, N.R., 1995. *Intelligent Agents: Theory and Practice.* Manchester: Department of Computing, University Of Manchester.

Xiang, X., Kennedy, R. and Madey, G., 2005. *Verification and Validation of Agent-based Scientific Simulation Models.* Notre Dame: Department of Computer Science and Engineering, University of Notre Dame.

Xu, J., Gao, Y., and Madey, G., 2003. *A Docking Experiment: Swarm and RePast for Social Network Modeling.* Notre Dame: Department of Computer Science and Engineering, University of Notre Dame.

Appendix A

DECarb

Decarb's Front-end Screenshots

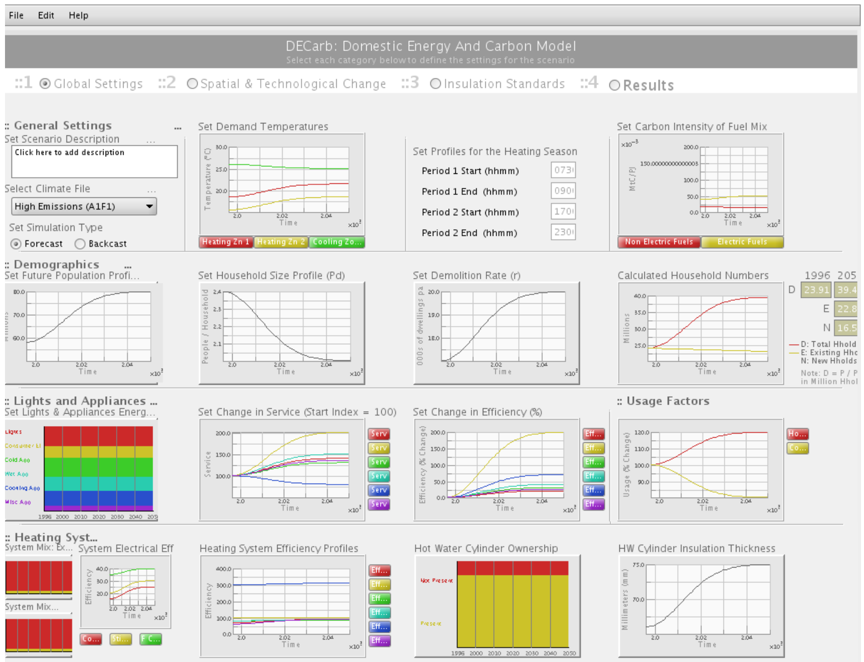

Figure 36: Screen 1 of DECarb

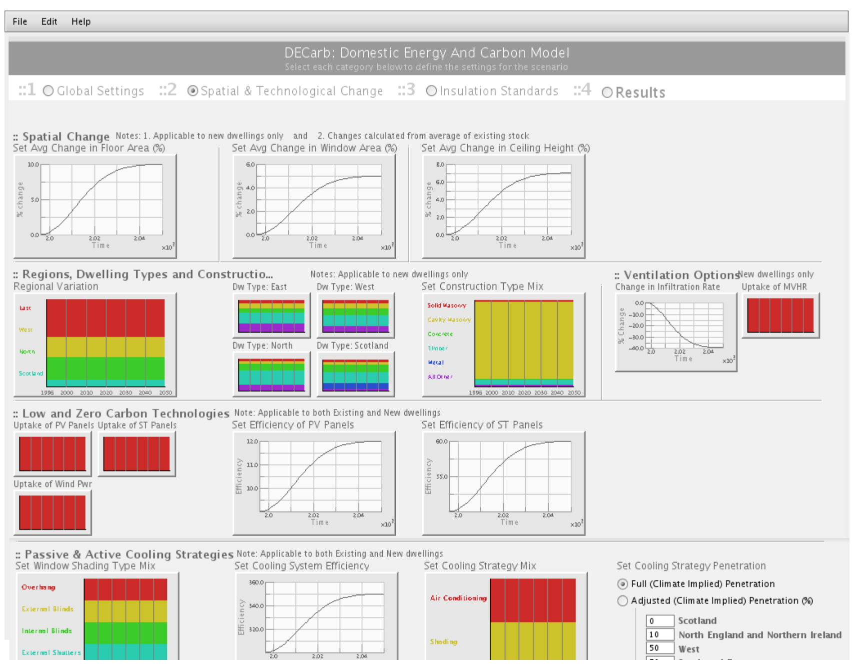

Figure 37: Screen 2 of DECarb

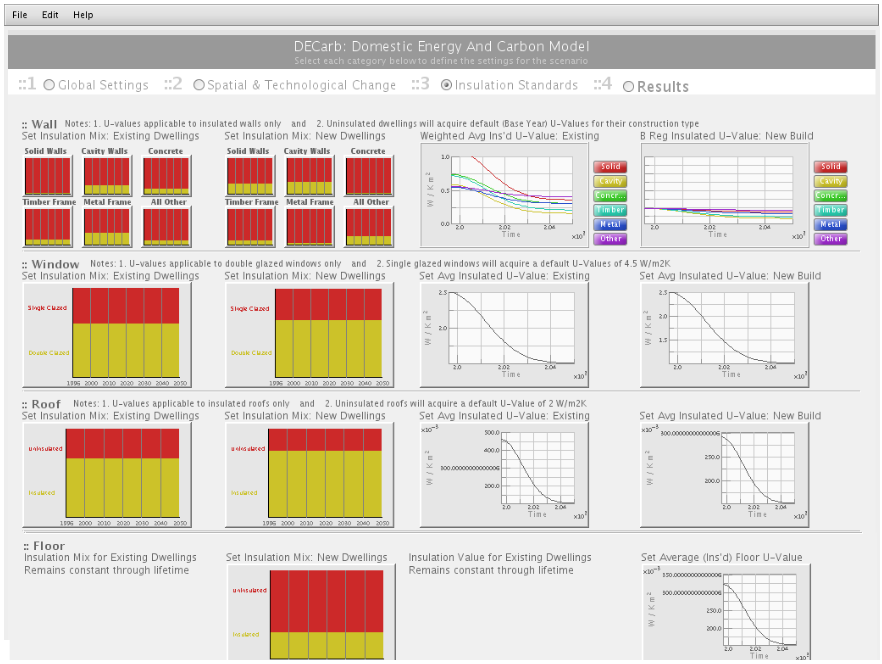

Figure 38: Screen 3 of DECarb

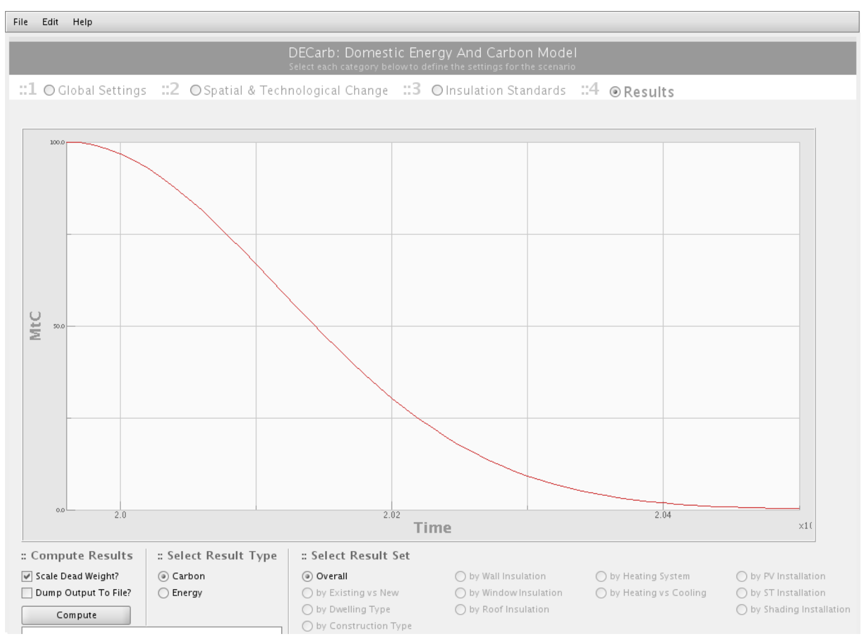

Figure 39: Screen 4 of DECarb

.man files

.man files can be used to specify a scenario in DECarb. The .man files used for both the back-cast scenario and the 40% House scenario are too large to include in this document and can be found on attached CD.

These files are all that are needed if any scenario is to be re-run. A .man file can be loaded in DECarb by going to File -> Load.

Formatting DECarb Output with Knoda

The implementation of the BREDEM algorithm used in DECarb to predict carbon emissions and energy consumption, does not produce output in a useful form. Huge quantities of data is produced in text files, and it requires analysis to turn it into a useful form.

To this end a piece of software was required to insert text files into a Database. Natarajan used Microsoft Access to extract figures from data in the original implementation, this was not an option, due to the constraints imposed by using Linux. A free piece of database software called Knoda (available from knoda.org) was used instead.

Given a database, Knoda allows the user to input data directly from text files, without any additional processing. Once the average data was contained in a database, it was a simple matter of writing queries in order to extract useful data.

Appendix B

Formulas

Getting the distribution of agents correct – Method 2

As demonstrated in figure 12, stock transformation method 1 was not accurate enough. So rather than start again from scratch, we stepped through method 1 to see exactly where the deviation from expected was taking place, and a major flaw was discovered.

Figure 40 is a proof that method 1 is inaccurate. The problem with it was that the constant value of U_i did not take into account the fact that Y gets progressively smaller.

Figure 41 gives a simpler example as to why the method did not work. Y_2 = 0.3 as expected, but on the next step we are taking 0.25 of Y_2, rather than 0.25 of Y_1, which means not as many households change group as intended and Y_3 = 0.225 rather than 0.2.

This left two possible solutions. At every step, the fraction of agents that should obtain double-glazing could have been altered. For example $\forall\, Y_i \exists X \;\; s.t. \;\; X \cdot Y_i = U_i \cdot Y_1$ and this new X could have been used as a new value of U_i. But this solution would have required additional calculation at every step of the model.

$$
\begin{aligned}
& Y_1 \\
Y_2 &= Y_1 - (Y_1 \cdot U_i) \\
Y_3 &= Y_2 - (Y_2 \cdot U_i) \\
& . \\
& . \\
Y_i &= Y_{i-1} - (Y_i \cdot U_i) \\
& . \\
& . \\
Y_N &= Y_{N-1} - (Y_{N-1} \cdot U_i)
\end{aligned}
$$

$$\forall\, i > 1$$

$$Y_i < Y_1 \;\; therefore \;\; (Y_i \cdot U_i) < (Y_1 \cdot U_i)$$

$$(Y_1 \cdot U_i) + (Y_2 \cdot U_i) + \ldots + (Y_N \cdot U_i) < Y_1 \cdot (N \cdot U_i)$$

Figure 40: Proof of why method 1 is inaccurate

$$N=3$$
$$Y_3=0.2$$
$$P_2=0.5$$
$$U_i=0.25$$

$$Y_1=0.4$$
$$Y_2 = 0.4-(0.4*0.25) = 0.3$$
$$Y_3 = 0.3-(\mathbf{0.3}*0.25) = 0.225$$

Figure 41: Example showing inaccuracies with method 1

Compound Interest Formula

$$FV=PV \cdot (1 + I)^n$$

Where I is the interest rate, FV and PV are future and present value of a sum, n represents the number of periods

Appendix C

Source Code

Source code and simple running instructions can be found on the attached CD. They will also be hosted at www.bath.ac.uk/~lje21/Proj until 31st May 2008.

Appendix D

Graphs

Graphs required to back-up a point, but not considered essential enough to interrupt the flow of text are included in the this appendix.

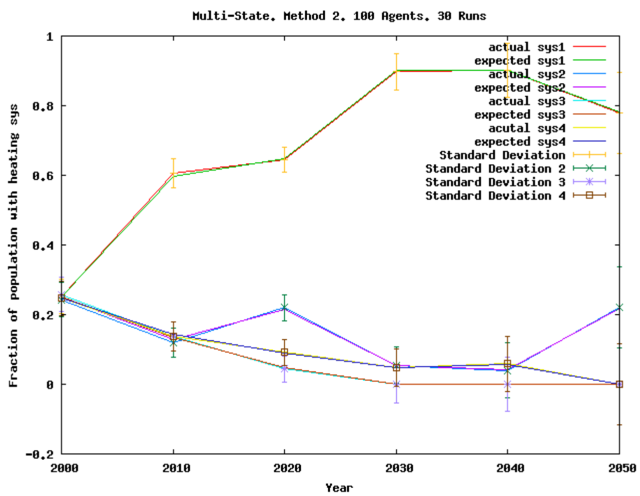

Figure 42: 4-Attribute Tracing

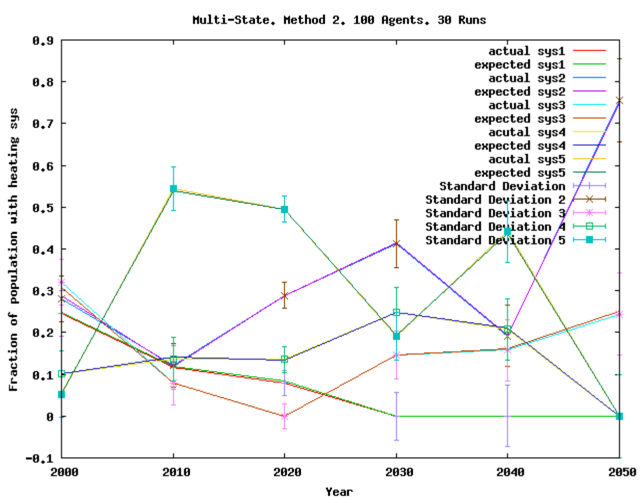

Figure 43: 5 - Attribute Tracing

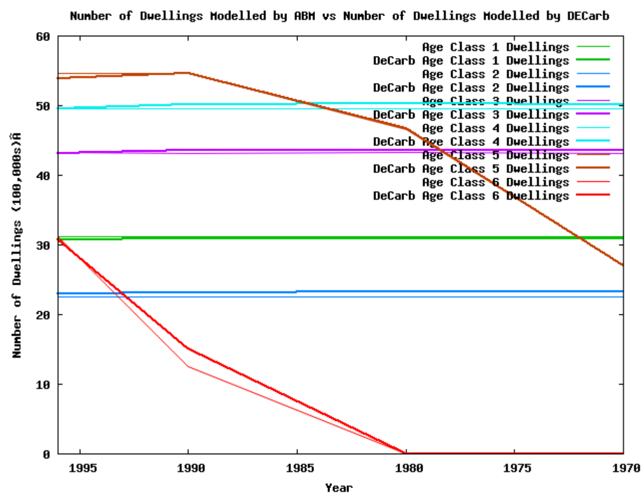

Figure 44: Total Dwellings Modelled by ABM vs Modelled by DECarb

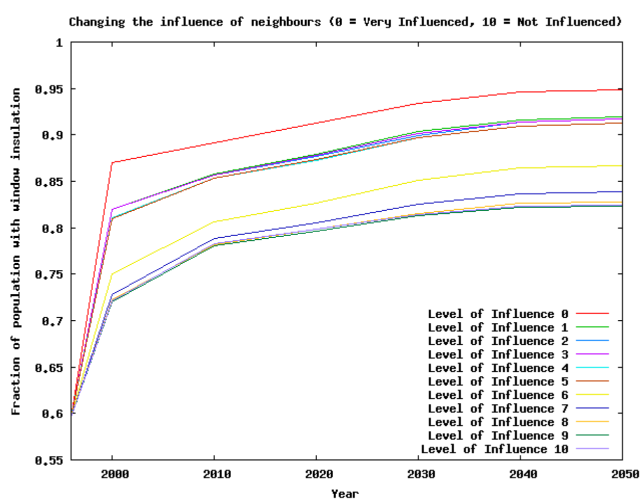

Figure 45: Parameter sweep on how influential neighbours are with a 1 year government scheme in 2000

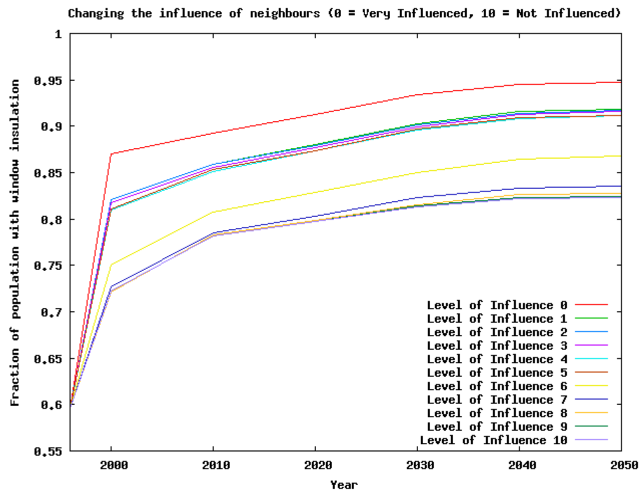

Figure 46: Parameter sweep on how influential neighbours are with a 2 year government scheme in 2000

Appendix E

Raw Results Calculation

Calculation of deviation from actual in back-cast results

Using actual data on energy consumption from each of the years, the deviation of the ABM's results from the actual can be calculated. 3 periods of under-estimation are compensated for by the figure estimated for 1990, which is a vast over-estimate.

Table 13: Energy Consumption Deviation from Actual

Energy Consumed	1996	1990	1980	1970
Modelled	1985	1998	1654	1470
Actual	2014	1706	1668	1544
Difference	**-1.44%**	**17.12%**	**-0.84%**	**-4.70%**

The data for carbon emissions shows a similar pattern. No actual figures are available for this period by Natarajan & Levermore (2007a) use data calculated from the Domestic Energy Fact File (DEFF).The results output by the ABM are under-estimates with the exception of 1990, which in this case is only a slight over-estimate. The break-down of this data created by DECarb is not currently available, but from Natarajan & Levermore it is possible to derive the figures shown in table 8.

Table 14: Carbon Emissions Deviation from the DEFF Calculations

Carbon Emissions	1996	1990	1980	1970
Modelled	38.4	43.6	43.97	52.17
DEFF	40.7	42.4	46.6	53.3
Difference	**-5.65%**	**2.83%**	**-5.64%**	**-2.12%**